Proceedings of the Ninth Scandinavian Conference on Artificial Intelligence (SCAI 2006)

Helsinki University of Technology
Espoo, Finland, October 25-27, 2006

Timo Honkela, Tapani Raiko, Jukka Kortela, and Harri Valpola (eds.)

Using the Self-Organizing Map for Measuring Interdisciplinary Research

Henrik Bruun*
Institutions and Social Mechanisms
(IASM), University of Turku
Henrik.bruun@utu.fi

Sampsa Laine†
†Data Rangers Oy, Helsinki, Finland
Sampsa.laine@datarangers.fi

Abstract

The prevalence and role of interdisciplinary research has been debated both among scholars and policy makers. Unfortunately, there is a lack of good tools for measuring how common such research is, and how important it is in the dynamics of science. In this paper we set out to develop a new tool for the measurement of interdisciplinary research; a tool that would use the self-organizing map (SOM) as its technological basis, and the WEBSOM method for a contents analysis of documents. Our focus is on the requirements that such an analytical tool would have to fulfil.

1. Introduction

Interdisciplinarity research is in many ways a contested phenomenon among scholars studying science. There is uncertainty about its prevalence; its basic forms; its significance for science; its quality; its relevance for society; and how it should be implemented. In this paper, we focus on the first two topics, the prevalence and significance of interdisciplinary research. There are presently two major ways of studying interdisciplinary research: through case studies and through scientometric measurement. The first methodology includes studies of: the formation of disciplines; the activity of research groups or institutions; trends in specific research areas; and the development of innovations. It also includes reviews of research programs and organizations in various areas. The second methodology covers surveys and bibliometric studies of interdisciplinary activity.

Taken together, the research based on case studies and scientometric methodology indicate that interdisciplinary research is indeed a common phenomenon, and that it is of great significance for the dynamics of science. Some of the research also suggests that there is a trend toward an increase in the crossing of disciplinary boundaries. However, there is no convincing method for demonstrating these things numerically. The case studies do not aim at measurements, and the scientometric methods have some problematic limitations. As an improvement to contemporary methodologies, we propose that the self-organizing map (SOM) is used as the basis for measuring interdisciplinarity. In what follows, we outline the requirements of such a tool.

The paper is structured as follows. In the next section, we review some of the reasons for why interdisciplinary research is thought to be common and important, respectively rare and of marginal importance. In section three, we review the methods by which various scholars have tried to establish the prevalence and trends of interdisciplinary research. We also discuss the limits of those methods, focusing particularly on the existing bibliometric methods. In section four we propose that the self-organizing map (SOM) could be used to achieve a higher degree of validity in the analysis. We outline some of the principles for how the SOM could be used for this. Section five concludes the paper with a discussion about some of the challenges for further development of the SOM methodology discussed here.

2. The significance and prevalence of interdisciplinary research

Few issues are as contested as the significance and prevalence of interdisciplinary research. Opinions among scholars vary from Alan Leshner’s claim (quoted in Johansson 2004, 27) that “disciplinary science has died” and that “most major advancements involve multiple disciplines,” to Peter Weingart’s (2000, 26-27) contention that scientific innovation takes place “primarily within disciplines, and it is judged by disciplinary criteria of validation.” The two scholars should know what they talk about. Leshner is the CEO of the American Association for the Advancement of Science (AAAS), the world’s largest science organization and the publisher of *Science*. Weingart, in his turn, has been the Director of University of Bielefeld’s s well known Center for Interdisciplinary Studies (ZiF). Still, Leshner and Weingart have contradictory views on the significance and prevalence of interdisciplinary research. For Leshner it is the engine of science; for Weingart it is merely an ideal, encouraged by science funders, but poorly mediated to the practice of doing science.

Both poles of opinion have their followers in the scientific community. Leshner receives support from scholars such as Klein (1993, 187), who maintains that disciplines permeate each other, and that “permeation occurs across the distance of discipline, from frontier to core.” Similarly, Gibbons and his colleagues (1994, 5) argue that final solutions in modern scientific knowledge production (what they call Mode 2) “will normally be beyond that of any single contributing discipline.” Stefik and Stefik (2004) agree, and point out that collaboration across disciplinary boundaries is particularly important in radically new research, or what they denote breakthrough research.

Gibbons and the Stefiks are primarily talking about the natural, life and medical sciences, and about technological research. Wallerstein and his colleagues bring the same argument to the social sciences. They write that "the tripartite division between the natural sciences, the social sciences, and the humanities is no longer as self-evident as it once seemed," and also contend that social science disciplines have become increasingly heterogeneous (Gullbenkian Commission 1996, 69; see also Dogan and Pahre 1990, and Camic and Joas 2004 for similar arguments). What about the humanities, then? Klein's (1996, 133-172) study of literary studies shows that the subject of interdisciplinarity has recurred over and over again within that field. Moran (2002) comes to a similar conclusion in his analysis of literary studies: "it has never been a 'pure' discipline but a hotchpotch of contending aesthetic, theoretical and scientific discourses."

Contesting all these claims, however, Abbot (2002, 227) makes the case that "disciplines work surprisingly well as a cultural system" and predicts that "in fifty years, we will probably have a disciplinary system recognizably descended from the present one." Scholars who reason along these lines agree with Weingart about the primacy of disciplines. Some even question the practical possibility of interdisciplinary work, arguing that "outsiders cannot properly practice an intellectual discipline just as foreigners find it difficult to assimilate into a national culture" (Bauer 1990, 114). Several studies show that interdisciplinary research can be difficult as a result of disciplinary boundaries (e.g., Wallén 1981; Messing 1996; Rabinow 1996; Jeffrey 2003).

Practitioners of interdisciplinary research often complain about the scepticism towards, and resistance to, interdisciplinarity within academia (see Klein 1990). Yet it is difficult to find public stances against interdisciplinary work, perhaps because of the fact that interdisciplinarity expresses a scientific norm shared by most scientists, that of originality. As Weingart (2000, 31) points out, "public pronouncements of scientists will regularly reflect an openness towards interdisciplinary research, since it is concurrent with the value of originality. And they will, conversely, contain denials of 'discipline-mindedness' except in connection with efforts to secure an innovation." However, this does not mean that they endorse interdisciplinarity in their operations and practice. As is well known, the scientific norms do not organize practice in the way assumed by Merton (1973). Weingart's (ibid.) conclusion is that "declarations by scientists about the desirability of interdisciplinary research cannot be taken at face value."

3. Empirical study of interdisciplinary research

For a long time, knowledge about interdisciplinarity relied mainly on accounts of first hand experience and case studies. This was problematic, because such studies cannot by themselves provide a basis for generalizations. When aggregated, however, the sheer number of anecdotes and case studies of interdisciplinary endeavors suggests that interdisciplinary research is indeed possible and not that uncommon. The anecdotal and case study -based literature includes examples of interdisciplinary research in a large number of fields, including landscape research (Tress, Tress et al. 2003), space research (Bonnet 1999), and materials science (Cahn 2000); research institutions (Hollingsworth and Hollingsworth 2000; Maasen 2000; Scerri 2000; Höyssä, Bruun et al. 2004; Stefik and Stefik 2004); and large scale mission-oriented projects (Hughes 1998). This literature shows that a great number of scientific and technological breakthroughs have had their background in interdisciplinary modes of research (for more examples, see Rabinow 1996; Hughes 1998; Miettinen, Lehenkari et al. 1999; Hollingsworth and Hollingsworth 2000; Wilmut, Campbell et al. 2000; Stefik and Stefik 2004; Langlais, Bruun et al. 2004).

Historical studies of the emergence of new research fields, disciplines, and technologies, provide similar evidence. Consider, for instance, the influence of physics in molecular biology (Morange 1998); the interaction between physics and chemistry in research on high temperature superconductivity (Nowotny and Felt 1997); the involvement of a large number of disciplines in artificial intelligence, cognitive science and neuroscience (McCorduck 2004); and the interaction between history and sociology in historical sociology and sociological history (Dogan and Pahre 1990; Gulbenkian Commission on the Restructuring of the Social Sciences 1996). This historical evidence suggests that interdisciplinary communication and interaction often plays a key role in the emergence of new research fields, that is, in scientific renewal and development.

Yet another type of evidence consists of historical and contemporary *institutional* signs of interdisciplinary research activity, such as the changing emphasis on interdisciplinary research in policies and principles for organization, and the establishment of interdisciplinary programs, university departments, centers, institutes, networks, etc. The great number of previous and contemporary interdisciplinary institutions, reported by Klein (1996) and others (see chapters in Salter and Hearn 1996; Cunningham 1999; Roy 2000; Weingart and Stehr 2000), indicates that interdisciplinary research is indeed a widespread phenomenon in academia. This evidence is suggestive but not conclusive, however. Institutional mappings may fail to identify the real character of the activities within those institutions. In-depth studies of research programs that characterize themselves as interdisciplinary may reveal that they are multidisciplinary rather than interdisciplinary, or just fragmented in completely unconnected disciplinary work, as predicted by Weingart.

At the same time, it should be observed that much interdisciplinary work is going on within the framework of the traditional, disciplinary department structure of universities (Dogan and Pahre 1990; Schild and Sörlin 2002). Such activities are difficult to register if attention is given to interdisciplinary institutions only.

A fifth type of information about the role of interdisciplinary research comes from more comprehensive studies of the behavior and experiences of scholars. The methods used in these studies range from surveys and interviews to bibliometric publication counts. Common for them all is that they base their evidence on large samples of scholars or scholarly outcome (in contrast to, for instance, case studies). Morrison et al. (2003) did a survey study of 144 academic staff across 15 disciplines in the Faculty of Science at a New Zealand university. They found that over 85% of the respondents were involved in one or more collaboration projects, but that most of the projects were disciplinary in orientation. Only 6% of all projects were interdisciplinary. On the other hand, over half of the staff (56%) considered interdisciplinary collaboration to be important.

A Korean study gave a different picture. Song (2003) analyzed 4,163 proposals submitted to the Korea science and engineering foundation (KOSEF) in 2000 and 2001. The applications represented twelve fields of research within science and engineering. KOSEF requests researchers to indicate primary, secondary, and tertiary research fields and to estimate individual weights of each field in their proposals. Song found that 35.8% of individual research proposals and 54.6% of collaborative proposals were interdisciplinary.[1] He also found that the average weight of non-primary disciplines was 11.3% in individual research plans and 19.4% in collaborative plans. The degree of interdisciplinarity varied across the twelve fields. Still, Song' conclusion is that interdisciplinary research "already prevalent in Korea." Another conclusion is that interdisciplinary research is more common in collaborative research.

Bruun et al.'s (2005) study of applications for funding to the Academy of Finland, and the funding decisions of that organization, produced results more consistent with those of Song than with those of Morrison et al. Like Song, Bruun and his colleagues used applications as empirical material, but their method for categorization was different. They classified the applications on the basis of a qualitative analysis of their contents. Bruun et al. found that more than 40% of a sample of 324 successful research applications proposed to do interdisciplinary (in the generic sense) research. In a closer scrutiny of 266 of those applications, they found that 17% of them promised to do multidisciplinary research, while 26% promised more integrated, or interdisciplinary (in the specific sense), research. The integrative approach was thus more common than the multidisciplinary approach, and approximately one fourth of all the General Research Grant funding was allocated to such projects in 1997 and 2004.

Interdisciplinarity can also be assessed by analyzing publication activity. Dutch researchers have studied interdisciplinarity by creating publication-based research profiles for institutions such as research programs or institutes. Rinia et al. (2001), for instance, studied a sample of 17,760 publications from 185 physics research programs in the Netherlands. They used the ISI (Institute of Science Information) journal classification to categorize all publications. The publication categories were then used to create a research profile for each program. The research profile tells us how the publications of the program were distributed across (sub)fields. In this case, the more papers published in non-physics journals, the more interdisciplinary is the research profile. With this operationalization of interdisciplinarity, the average degree of interdisciplinarity of a physics program was 36%. More than a third of publications were thus published in non-physics journals.

Another research profile study, of a well known Nutrition and Food Research institute in the Netherlands, analyzed 1,395 publications published by institute researchers in 1987-1996 (van Raan and van Leeuwen 2002). The methodology for categorizing publications was the same as above. This time, however, the institute's output was broken down into research fields rather than aggregated to a number. The study showed that the institute's output was highly interdisciplinary in the sense that it was distributed across a large number of fields, and that, more significantly, it succeeded in having a high impact in twelve different fields. This does not tell us much about the interdisciplinarity of actual research activities, of course, but illustrates that multidisciplinary research environments can produce good quality work.

Bibliometric studies give further confirmation of the idea that the significance of interdisciplinarity varies across research fields. Qin et al. (1997) studied 846 scientific research papers that were randomly selected from the Science Citation Index for publications in 1992. They measured a paper's degree of interdisciplinarity with the number of disciplines represented by journals cited in its bibliography. Ulrich's International Periodicals Directory was used to obtain category information for the journals. The study shows that 76% of the papers were written by more than one author. The degree of interdisciplinarity (the number of cited disciplines) ranged from 1.78 in mathematics to 5.18 in agriculture. One third of the collaborative papers were produced by authors from departments in two different disciplines. Two thirds of collaborations were thus between scholars from the same discipline, or, to be more specific, from departments with the same disciplinary label. This did not, however, necessarily mean that work was strictly disciplinary. "Within-disciplinary" collaborative projects frequently cited journals from other fields. Qin et al.'s conclusion is that "*limited* scientists-scientist (in terms of affiliation) interaction still can involve

[1] The exactness of the numbers appears strange to us, considering the method that was used. Yet, we did not to manipulate the numbers reported by Song.

extensive scientist-information interaction." (p. 913, our italics)

In sum, then, there are many ways of studying interdisciplinarity in research. All methods have their strengths and weaknesses. The historical case studies and reviews of institutions give a lot of detailed information, but are limited in that they do not allow us to compare the prevalence or significance of interdisciplinary activity with that of disciplinary activity. The questionnaire-studies teach us a lot about attitudes or behaviour in one particular institutional context, but say little about the generality of the findings. Also, questionnaire-studies give little information about trends, unless repeated several times. The study of research applications teaches us a lot about the researchers' intents, but little about actual implementation. Again, it is unclear what generality the findings have, unless compared with other, similar studies. Unfortunately there have been no large-scale, comparative studies of interdisciplinarity, based on questionnaires or applications.

The easiest way of doing large-scale research on interdisciplinarity is to use bibliometric tools. The problem with the existing tools is, however, that they use metadata only, which implies that their operationalizations of interdisciplinarity are rather crude. Bibliometric metadata consists of the information that is attached to the publications that are studied: name of author, organization and country of author, key words, name of journal, etc. As mentioned above, a publication is considered to be interdisciplinary if, for instance, the organization of the author (e.g. a university department) belongs to a different disciplinary categorization than the journal in which the article in question is published. Thus a researcher in a biology department who publishes a paper in a physics journal is considered to have produced an interdisciplinary product, per definition. The problem here is that this presumes that there is a correspondence between the organizational category and the research of the scientist. In the example above, such assumptions are problematic if the author is a physicist employed by a biology department. A normal, disciplinary paper is then taken to be interdisciplinary. In other cases, the mistake can go in the other direction: an interdisciplinary product is perceived as disciplinary.

It can of course be argued that the problem of non-correspondence is cancelled when data is aggregated. Cases like the philosophical paper, produced at a technical university, and published in a sociological journal, are then considered to be exceptions—noise in the data. But it is precisely this issue that is under investigation here. Are they exceptions? And even if they were exceptions, what is their significance? Is a new area of research more likely to grow from the work of a biologist publishing in a biologist journal, or from a computer scientist publishing in a biologist journal? In the rest of this paper, we will focus on the development of a bibliometrical tool that would be more sensitive to the actual contents of publications, and that would allow us to study not only the prevalence of interdisciplinary publications, but also their significance.

4. A New Tool for Measuring Interdisciplinarity

We envision a bibliometric tool that automatically analyses the contents of scientific publications, and that compares those contents with regard to similarity. The Self-Organizing Map (SOM) could constitute the basis of such a tool (Kohonen 2001). The SOM is a technique for a two-dimensional clustering of high-dimensional data, developed by Teuvo Kohonen in the 1980s. The SOM has been applied for a number of purposes, including the analysis of publication contents. In the latter case, the analysis uses word-use as a proxy for content. Thus, publications that use words in similar ways are considered to have similar contents. There are a few encouraging studies of the validity of such an approach (e.g., Pöllä et al. 2006), but more testing is needed. Although the SOM-technique was developed some twenty years ago, its application for research purposes is still at an early stage, particularly when it comes to content analysis. In the following, we rely on the WEBSOM method that was developed by Honkela et al. (1997) for the purpose of contents analysis (see also Kaski et al. 1998; Kohonen et al. 2000; Lagus et al.2004).

The procedural steps of the WEBSOM method are as follows. The first task is to transform the texts into numerical data. This is achieved by treating them as high-dimensional data units, and by transforming them to vectors. The vectors are then used as input data in the SOM-analysis. How is this done in practice? One starts by defining each publication as a primary data unit, and the words used in them as a dimension for a property. A text with 500 different words, then, has properties in 500 dimensions. These properties can vary in each dimension, which means that the dimensions are, in effect, variables. To illustrate, in the present text, the word "SOM" constitutes one of its dimensions. If the text uses this word thirty-four times, the text has the property of having 34 instances of "SOM". A vector can then be formed on the basis of the word-use in the text: each word constitutes a vector-dimension, and the number of times that the word is used constitutes the value of that dimension (Lagus 1997). When several texts are analysed, a data matrix must be formed of all the words used in the complete body of sample texts. Let us say that 100 texts are analyzed, and that they use 2,200 words altogether. Then each text is transformed to a vector with values in all of these 2,200 dimensions. In the case of a text with 500 words only, 1,700 dimensions get the value of zero.

The next step is to pre-process the vector data. We will not go into the details here, but only mention two things that need consideration. First, some kind of normalization is needed. Second, words that are very common create background noise in the data, and need to be omitted. There are various techniques for this, such as comparing the sample texts with some neutral body of texts, and then omitting words that occur frequently in both. A second method is to use word-lists for the exclusion of frequent words. A third method is to use some technique for the determination of superfluous words, such as entropy-based techniques that omit words that are evenly distributed across the texts. These procedures reduce the number of words available to the analysis, or in technical terms, the number of dimensions of the sample text vectors. In the example above, the number of dimension may be reduced from 2,200 to 1,300. Having pre-processed the data, the SOM iteration can be initialized.

The SOM consists of a two-dimensional map of evenly and symmetrically distributed cells in which each cell is represented by a vector. In contrast to the input vectors (the sample text vectors), the vectors on the map are called prototype vectors. The prototype vectors have equally many dimensions as the input vectors, 1,300 in our example, and may be set with random values. The purpose is now to map the information in the input data onto the SOM, so that the vectors on the map will reflect the topological organization of the data in the input space. This is more challenging than it sounds, because in the input space the topological organization is very complex, distributed in 1,300 dimensions. In the SOM map that will result, however, the same information is organized in two dimensions only, which allows visual inspection and comparison for human beings (who cannot compare high-dimensional data).

To make the significance of this transformation more obvious, we can think of each text as a profile of words and word-usage. In order to compare the texts on the basis of these profiles, we would have to take the profiles as a whole into consideration. In practice this is impossible for human beings. Normally, when people say that two texts are similar, they base their judgement on a small number of properties, such as the plot or the style of writing. People use these holistic properties to reduce the number of dimensions to be compared. Unfortunately, contemporary automatized content analysis techniques cannot make such holistic, or abstract, assessments of texts, and therefore have to use other techniques, such as the word-count proposed here. Is word count also a holistic assessment technique, or is it reductionistic, considering that words are very small units in the text? If only one or a few words were used, the technique would be highly reductionistic (claiming, for instance, that similarity can be analysed on the basis of a few key words). The SOM technique, however, uses the word-profile to get a *different* holistic assessment of the text than the human one: the profile of words and word-usage.

The mapping of input data onto the SOM works as follows. One of the input vectors is compared with all of the prototype vectors in terms of distance. This can be done with normal vector mathematics, which allows the measurement of distances between high-dimensional vectors. The prototype vector that is closest to the input vector is called the winner. The winner is adjusted according to the SOM algorithm so as to be a bit closer to the input vector in question. In addition to the winner, some of its neighbours are also changed, but less so. These adjustments are called learning, and the rate with which adjustment is performed is called the learning rate. The learning rate and the size of the neighbourhood that learns are parameters that need to be determined before the mapping. When the learning has been accomplished on the map for the first input vector, a new input vectors is selected, and the same process is repeated. The iteration continues until the map stabilizes, that is, until the introduction of new input data has little effect on the prototype vectors. There are two technical reasons for such stabilization: the learning rate is reduced and the neighbourhood size shrinks over time. Both of these are parameters that can be varied.

The input data has now been coded onto the map. The next step is to do a new mapping of input data. This time, however, the map does not change. The purpose of the second mapping is not to map information, but to map representation. The second mapping determines how prototype vectors represent input vectors. For instance, if text A was part of the input data, and d is the prototype vector that is closest to it, then d represents A. The SOM is thereby transformed to a representative map in which cells on the map represent one or several data units in the input space. This representative property is illustrated by labelling the cells according to the input vector that they represent, for instance with a few words from the title of the sample publication in question. The outcome is a two-dimensional map in which the labels of cells show how the input data is distributed (see Figure 1). There are many complexities in the labelling. If the SOM-analysis is done with a large number of sample texts, some kind of automatic labelling is needed. Labels need to be short and informative, so full titles cannot be used. Furthermore, if the input material is large, each cell on the SOM will represent several texts, which makes the task of labelling even more challenging.

After labelling, the actual analysis of the map can begin. What kind of information does it contain? As said before, its primary function is to give a two-dimensional representation of the topology of the input space. Texts will thus not be evenly distributed in the SOM, but will be grouped according to similarity in content (word-usage profile). Texts with similar contents will cluster in specific regions of the map. This visual illustration of clustering is improved by the fact that the map is non-linear: vector distances do not correspond to metric distances. A distance of three cells on the map can represent very different vector distances, depending on the values of the vectors. Visual aids have been developed to correct for the non-linearity. Thus, in many SOMs, large distances are illustrated with dark shades of gray and small distances with light shades of gray. Lightly coloured regions thus represent input data that is closely related, for instance in terms of content, while dark regions signify distance (see Figure 1). Using a three-dimensional metaphor, one can think of the light regions as valleys in which you can fly through the air between locations, and the dark areas as hills that have to be climbed when moving from location to location.

The SOM allows us to analyze the topology of contents in the samples texts. On the basis of the hierarchical model of scientific knowledge production, discussed in section 2, we would expect a SOM based on a large and heterogeneous body of scientific articles to have a disciplinary organization. There should in other words be a clear clustering around research fields, and these clusters should themselves be clustered in a way that allows us to find the established disciplines at higher levels of the hierarchy. The rhizome-model, on the other hand, predicts that boundaries between research areas will be diffuse, and that the connections between lower level clusters (research fields) and higher level clusters (disciplines) are complex. In order to do this investigation, the SOM itself would have to be hierarchically organized. What we mean is that by clicking on a cell on the map, one should be able to get a new SOM of the documents represented by that cell. It would also be important to be able to select several cells, and to get a new SOM of all the publications represented by those cells. The latter is important to be able to check for artificially produced hierarchies (a hierarchical SOM may produce an impression of hierarchically organized science, unless we can check for this effect).

We have chosen the SOM for the analyses of this paper since the SOM is robust and understandable. According to Laine (2003) these are important properties of systems intended for non-mathematicians. Robustness allows versatile types of data to be analyzed without pre-treatment. The SOM is robust since it makes no assumptions of the probability distributions of the data. The algorithm tolerates e.g. outliers and non-linearities. The SOM is understandable as the training concept of the SOM is simple and can be explained to the common user in minutes. The user is not required to understand probability mathematics or linear algebra.

Visual analysis needs, of course, to be supplemented by numerical analysis. We would need values for the degree of clustering of any region selected on the map. One should also be able to analyse clustering on the basis of metadata. For instance, we may be interested in the degree of clustering of the publication of author A or institute I. These measures correspond to the measure of degree of interdisciplinarity (see previous section), but in reversed form. A high degree of clustering would indicate strong specialization, while a low degree of clustering suggests an interdisciplinary profile. The clustering of a researcher or an organization can be compared with the clustering of other researchers or organizations, to see for instance how the researcher's publication profile compares with those of other researchers in his field. Such comparisons lay the foundation for a relative measure of specialization respectively interdisciplinarity. The good thing with these measures is that they are not dependent on any operationalization of specialization or interdisciplinarity from the analyst's side, but are based on an analysis of the topology of contents. The task of the analyst is then to draw the line between specialization and interdisciplinarity, and to distinguish different categories of the two. There will probably be a continuum from very specialized to very broad.

Numerous other analyses are possible by using metadata to investigate distributions on the map. One could study the clustering of, for example, biology journals, sociology departments, or Finnish chemists. Comparisons of clustering could also be made between, for instance, psychology journals based in the US and Europe; Norwegian and Danish economists; Indian and Chinese information technology departments; or top-level and mediocre universities. One could also compare productive scientists with less productive scientists. Which group tends to be broader in its publication profile? Time is a particularly important type of metadata. If publications from different years are included, the SOM becomes, in effect, a time-space projection. The map as a whole does not represent the topology of contents at any particular year, but over a time continuum. With a large body of sample texts, we should thus be able follow how people, organizations, journals or whole nations move on the map over time. Do they become more specialized or more interdisciplinary? One should also be able to identify the origin of new clusterings. Are such origins to be found within older research fields or are they typically constituted by publications that move "outside the lines" (Salter and Hearn 1996). The temporal dimension allows us to analyse historical trends, and to project the future. Publication-based SOM-analysis could in other words be highly useful, both for development of theory and for policy purposes.

The usage of metadata to analyse the contents of the SOM requires both visual and numerical tools. Visual tools allow the analyst to see the distributions on the map, for example how the distribution of the publications of some particular organization moves over time. Such a tool would have to be advanced in the sense that it should not only show distributions in the year that is investigated, but also allow comparison with earlier distributions. This can be done, for instance, but letting historical and contemporary (the focus year) distributions be visible at the same time, but make distinctions by using different shades or colour for different years.

5. Discussion

This paper has proposed a new method for measuring interdisciplinarity. We started by discussing a range of opinions about the prevalence and significance of interdisciplinary research, and settled for a broad definition of the term. We then reviewed some of the empirical approaches to the study of interdisciplinarity. The review concluded that there are many different methodologies, and that all of them are limited either in scope or precision. Our own proposal, discussed in the previous section, uses the self-organized map and the WEBSOM method of contents analysis to achieve both scope and precision.

How does our concept relate to other efforts of mapping science? The idea of studying the structure of science is not new. Early studies focused on networks or clusters of authors, papers, or references. Later studies expanded the approach to co-word analysis, to achieve a greater semantic relevance. As visualization techniques have improved and computing capacity increased, samples have become larger and maps more complex. (Boyack et al. 2005) The focus in most previous mapping exercises, if not all of them, has been on *the map as the object of analysis*: maps show us the structure of science. Our proposal, however, uses a different approach. We do not define the map as the object of analysis, but rather as a space for meta-data behaviour. Meta-data behaviour is the true object of analysis, in our model. The map constitutes, instead, a space in which the meta-data events unfold. This puts some pressure on contemporary techniques for clustering and visualization. Our proposal goes beyond presently existing techniques, but may not be unrealistic in the near future.

At the core of the proposed concept is learning from data. According to Laine (2004), key features in such a system are supervised operation, robustness and understandability. A tool created for bibliometric analysis should allow the user to guide the search towards areas of interest (supervision), should operate well with heterogeneous text corpuses (robustness), and be usable to a wide range of users (understandability).

These three criteria apply to three levels of development: algorithm, software and service model development. On the algorithm level, we need to choose and develop statistical methods that comply with the three criteria mentioned above. The SOM is such an algorithm. On the software level, we need to create a human-computer –interface that is intuitive and leads the user towards results that he or she considers to be interesting. The SOM, as a visualization tool, provides the user with the grand view, and immediately allows the user to zoom into a given field of clustering of publications. In the present paper, we have proposed a number of solutions that are yet to be implemented. The underlying technologies exist for most of the propositions, for example, implemented in the public domain SOM-Toolbox created for Matlab (www.cis.hut.fi), but they need to be brought together in a new way. In the phase of implementation, it is important to remember that the concept has to serve real organizations. The needs of potential users, such as scholars interested in the mapping of science and policy makers, must be considered. We require an interdisciplinary project comprising domain experts of various academic and non-academic fields and computer science experts.

If such a project is successful, we gain a system that allows analysis of past developments, and to some extent, prognosis of future application areas. According to Hargadon (2003), breakthroughs happen when existing knowledge is combined in new ways, and a new team pursues the generated vision. The concept proposed in this paper allows new theoretical combinations to be identified, and their technological strength and potential for application to be estimated. What remains to be done is, in the terminology of Stefik and Stefik (2004), to follow the problem to its roots.

References

Abbott A. 2002. The Disciplines and the Future. The Future of the City of Intellect. Stanford, CA: Stanford University Press. p 205-230.

Bauer HH. 1990. Barriers against Interdisciplinarity - Implications for Studies of Science, Technology, and Society (Sts). Science Technology & Human Values 15(1):105-119.

Bonnet R. 1999. Space Research & Technology. An Example of Interdisciplinarity. In: Cunningham R, editor. Interdisciplinarity and the Organisation of Knowledge in Europe. A Conference organised by the Academia Europaea, Cambridge 24-26 September 1997. Luxembourg: Office for Official Publications of the European Communities. p 99-111.

Boyack, K., Klavans, R., Börner, K. 2005. Mapping the Backbone of Science. Scientometrics 64(3): 351-374.

Bruun H, Hukkinen J, Huutoniemi K, Klein JT. 2005. Promoting Interdisciplinary Research. The Case of the Academy of Finland. Publications of the Academy of Finland 8/2005. Helsinki.

Cahn R. 2000. Early Stirrings of Materials Science in Britain. In: Roy R, editor. The Interdisciplinary Imperative. Interactive Research and Education, Still an Elusive Goal in Academia. Lincoln, NE: Writers Club Press. p 85-90.

Camic C, Joas H. 2004. The Dialogical Turn. New Roles for Sociology in the Postdisciplinary Age. Lanham, MD: Rowman & Littlefield.

Cunningham R, editor. 1999. Interdisciplinarity and the Organisation of Knowledge in Europe. A Conference organised by the Academia Europaea, Cambridge 24-26 September 1997. Luxembourg: Office for Official Publications of the European Communities.

Dogan M, Pahre R. 1990. Creative Marginality. Innovation at the Intersections of Social Sciences. Boulder: Westview Press. x, 278 p. p.

Gibbons M, Limoges C, Nowotny H, Schwartzman S, Scott P, Trow M. 1997/1994. The New Production of Knowledge. The Dynamics of Science and Research in Contemporary Societies. London: SAGE Publications.

Gulbenkian Commission on the Restructuring of the Social Sciences. 1996. Open the Social Sciences. Report of the Gulbenkian Commission on the Restructuring of the Social Sciences. Stanford, Calif.: Stanford University Press. xv, 105 p. p.

Hargadon, A. 2003. How Breakthroughs Happen. The Surprising Truth About How Companies Innovate. Boston, Mass: Harvard Business School Press.

Hollingsworth R, Hollingsworth EJ. 2000. Major discoveries and biomedical research organizations: perspectives on interdisciplinarity, nurturing leadership, and integrated structure and cultures. In: Weingart P, Stehr N, editors. Practising Interdisciplinarity. Toronto: University of Toronto Press. p 215-244.

Honkela T, Kaski S, Lagus K, Kohonen T. 1997. WEBSOM--Self-Organizing Maps of Document Collections. Proceedings of WSOM'97, Workshop on Self-Organizing Maps, Espoo, Finland, June 4-6, pp.310-315. Espoo, Finland: Helsinki University of Technology.

Hughes TP. 1998. Rescuing Prometheus: Four Monumental Projects that Changed the Modern World. New York: Vintage. 372 p.

Höyssä M, Bruun H, Hukkinen J. 2004. The Co-Evolution of Social and Physical Infrastructure for Biotechnology Innovation in Turku, Finland. Research Policy 33(5):769-785.

Jeffrey P. 2003. Smoothing the waters: Observations on the process of cross-disciplinary research collaboration. Social Studies of Science 33(4):539-562.

Johansson F. 2004. The Medici Effect. Breakthrough Insights at the Intersection of Ideas, Concepts, and Cultures. Boston, Mass.: Harvard Business School Press. xiii, 207 p. p.

Kaski S, Honkela T, Lagus K, Kohonen T. 1998. WESBOM - Self-Organizing Maps of document Collections. Neurocomputing 21:101-117.

Klein JT. 1990. Interdisciplinarity. History, Theory, and Practice. Detroit: Wayne State University Press.

Klein J. 1993. Blurring, Cracking, and Crossing.Permeation and the Fracturing of Discipline. In: Messer-Davidov E, Shumway DR, Sylvan DJ, editors. Knowledges. Histroical and Critical Studies in Disciplinarity. Charlottesville and London: University Press of Virginia. p 185-211.

Klein JT. 1996. Crossing Boundaries. Knowledge, Disciplinarities, and Interdisciplinarities. Charlottesville and London: University Press of Virginia.

Kohonen T. 2001. Self-Organizing Maps. Berlin ; New York: Springer. xx, 501 p. p.

Kohonen T, Kaski S, Lagus K, Salojärvi J, Paatero V, Saarela A. 2000. Organization of a Massive Document Collection. IEEE Transactions on Neural Networks, Special Issue on Neural Networks for Data Mining and Knowledge Discovery 11(3):574-585.

Lagus K. 1997. Map of WSOM'97 Abstracts--Alternative Index. Proceedings of WSOM'97 Workshop on Self-Organizing Maps, Espoo, Finland, June 4-6, pp. 368-372. Espoo, Finland: Helsinki University of Technology.

Lagus K, Kaski S, Kohonen T. 2004. Mining Massive Document Collections by the WEBSOM Method. Information Sciences 163(3):135-156.

Laine, S. 2004, Using Visualization, Variable Selection and Feature Extraction to Learn from Industrial Data, Doctoral Thesis, Helsinki University of Technology, Finland, 56 pp.

Langlais R, Janasik N, Bruun H. 2004. Managing Knowledge Network Processes in the Commercialization of Science. Two Probiotica Discovery Processes in Finland and Sweden. Science Studies 17(1):34-56.

Maasen S. 2000. Inducing Interdisciplinarity: Irresistible Infliction? The Example of a Research Group at the Center for Interdisciplinary Research (ZiF), Bielefeld, Germany. In: Weingart P, Stehr N, editors. Practicing Interdisciplinarity. Toronto: University of Toronto Press. p 173-193.

McCorduck P. 2004. Machines Who Think. A Personal Inquiry into the History and Prospects of Artificial Intelligence. Natick, Mass.: A.K. Peters. xxx, 565 p.

Merton RK. 1973. The Sociology of Science: Theoretical and Empirical Investigations. Chicago: University of Chicago Press.

Messing K. 1996. Easier Said than Done. Biologists, egonomists, and sociologists collaborate to study health effects of the sexual division of labour. In: Salter L, Alison H, editors. Outside the Lines. Issues in Interdisciplinary Research. Montreal & Kingston: McGill-Queen's University Press. p 95-102.

Miettinen R, Lehenkari J, Hasu M, Hyvönen J. 1999. Osaaminen ja uuden luominen innovaatioverkoissa. Tutkimus kuudesta suomalaisesta innovaatiosta (Competence and the production of novelty in innovation networks. A study of six Finnish innovations). Helsinki: Sitra.

Moran J. 2002. Interdisciplinarity. London and New York: Routledge.

Morange M. 1998 [2000]. A History of Molecular Biology. Cambridge, MA, and London, UK: Harvard University Press.

Morrison P, Dobbie G, McDonald F. 2003. Research Collaboration among University Scientists. Higher Education Research and Development 22(3):276-295.

Nowotny H, Felt U. 1997 [2002]. After the breakthrough : the emergence of high-temperature superconductivity as a research field. Cambridge ; New York: Cambridge University Press. x, 210 p. p.

Pöllä, M, Honkela, T., Bruun, H., Russell, A. 2006. An Analysis of Interdisciplinary Text Corpora. Proceedings of The Ninth Scandinavian Conference on Artificial Intelligence (SCAI 2006), Helsinki on October 25-27.

Qin J, Lancaster FW, Allen B. 1997. Types and Levels of Collaboration in Interdisciplinary Research in the Sciences. Journal of the american Society for Information Science 48(10):893-916.

Rabinow P. 1996 [1997]. Making PCR. A Story of Biotechnology. Chicago and London: The University of Chicago Press.

Rinia EJ, van Leeuwen TN, van Vuren HG, van Raan AFJ. 2001. Influence of Interdisciplinarity on Peer-Review and Bibliometric Evaluations in Physics Research. Research Policy 30(3):357-361.

Roy R, editor. 2000. The Interdisciplinary Imperative. Interactive Research and Education, Still an Elusive Goal in Academia. San Jose: Writers Club Press.

Salter L, Hearn A. 1996. Outside the Lines. Issues in Interdisciplinary Research. Montreal & Kingston: McGill-Queen's University Press.

Scerri E. 2000. Interdisciplinary Research at the Caltech Beckman Institute. In: Weingart P, Stehr N, editors. Practicing Interdisciplinarity. Toronto: University of Toronto Press. p 194-214.

Schild I, Sörlin S. 2002. The Policy and Practice of Interdisciplinarity in the Swedish University Research System. Stockholm: SISTER. Report nr 2002:18.

Stefik M, Stefik B. 2004. Breakthrough! Stories and Strategies of Radical Innovation. Cambridge, Mass.: MIT Press. xiii, 294 p. p.

Song CH. 2003. Interdisciplinarity and knowledge inflow/outflow structure among science and engineering research in Korea. Scientometrics 58(1):129-141.

Tress B, Tress G, van der Valk A, Fry G, editors. 2003. Interdisciplinary and Transdisciplinary Landscape Studies: Potential and Limitations. Wageningen: Delta Series 2.

van Raan AFJ, van Leeuwen TN. 2002. Assessment of the scientific basis of interdisciplinary, applied research - Application of bibliometric methods in Nutrition and Food Research. Research Policy 31(4):611-632.

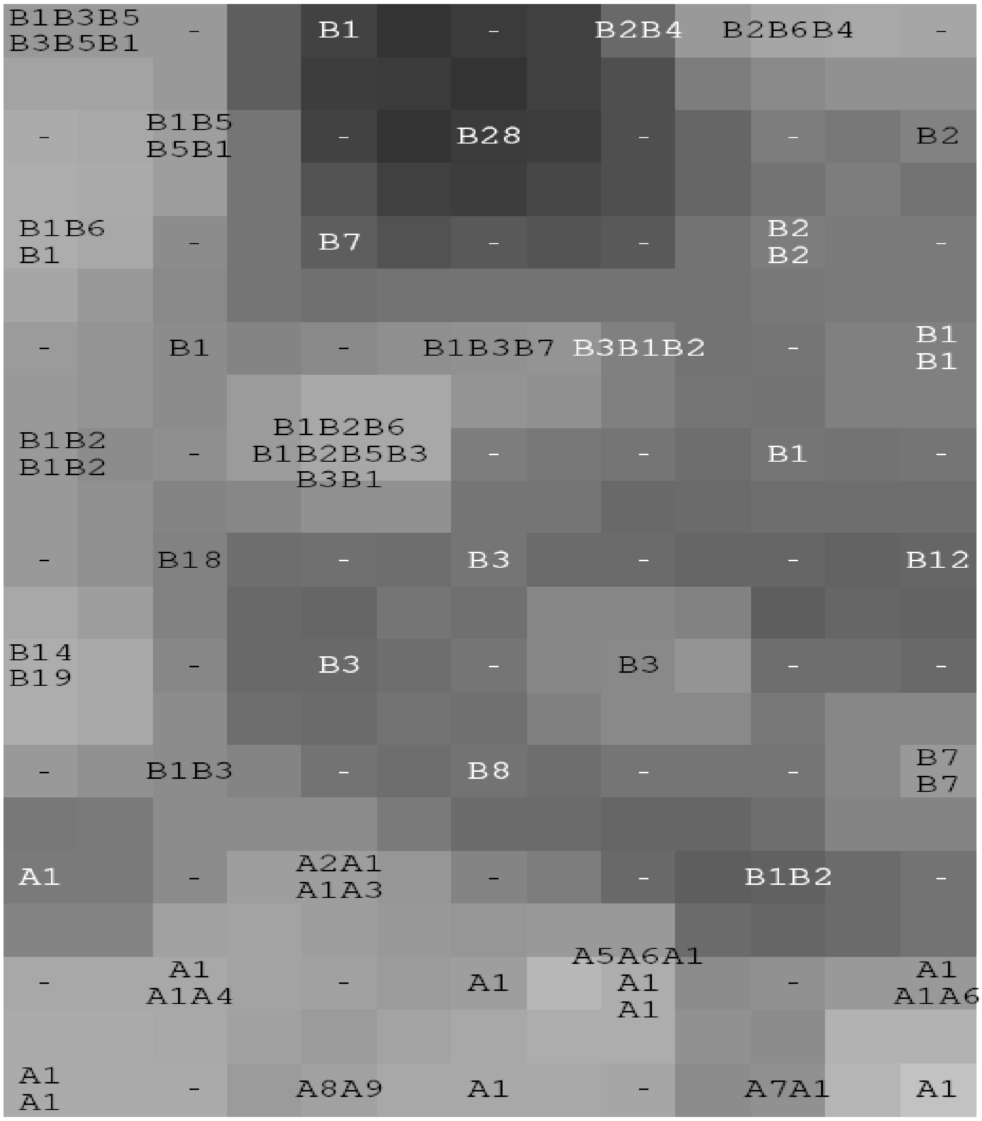
Wallén G. 1981. Tvärvetenskapliga problem i ett vetenskapsteoretiskt perspektiv (Interdisciplinary problems in the perspective of theory of science). Göteborg: Department of Theory of Science, Göteborg University. Report nr 130.

Weingart P. 2000. Interdisciplinarity. The Paradoxical Discourse. In: Weingart P, Stehr N, editors. Practicing Interdisciplinarity. Toronto: University of Toronto Press. p 25-41.

Weingart P, Stehr N, editors. 2000. Practicing Interdisciplinarity. Toronto: University of Toronto Press.

Wilmut I, Campbell K, Tudge C. 2000. The Second Creation. The Age of Biological Control by the Scientists who Cloned Dolly. London: Headline Book Publishing.

FIGURE 1: SOM of fifty-one publications (scientific papers). The symbols refer to the authors who produced the texts. Two disciplinary corpuses of text were investigated, A and B. In other words, A and B indicate the disciplinary orientation of the author. Numbers identify particular authors. B1 is thus author 1 from the disciplinary context of B. A second number identifies a particular text (in cases when individual authors have produced several texts). Thus, B28 refers to text 8 by author B2. This map is a draft version: the two number-system has not been used consistently (some of the texts by B2 are indicated with B2 only). In the final map, the two-number system will be omitted, since the identity of particular texts is not important for the readership. A few words need to be said about the colouring, too. Light areas indicate that publications are close to each other, even if the distance may be (relatively) long on the map. Dark areas indicate longer-than-metric distances. When investigating the distribution of text corpuses A and B, we see that they occupy distinct regions on the SOM. This seems to verify that the methodology is able to make adequate categorizations on the basis of contents. An unpublished analysis of the distribution of texts by B1, suggested that there was a high correspondence between the groupings made by the SOM and those made by B1 himself. Source: Pölla et al. 2006.



Professional Ethics in Computing and Intelligent Systems

Gordana Dodig-Crnkovic
Mälardalen University
Department of Computer Science and Electronics
Box 883, 721 23 Västerås, Sweden
gordana.dodig-crnkovic@mdh.se

Abstract

Research and engineering have a decisive impact on the development of the society, providing not only the material artifacts, but also the ideas and other "tools of thought" used to conceptualize and relate to the world. Scientists and engineers are therefore required to take into consideration the welfare, safety and health of the public affected by their professional activities. Research and Engineering Ethics are highly relevant for the field of computing (with Intelligent Systems/AI as its subfield). Computing Ethics has thus been developed as a particular branch of Applied Ethics. By professional organizations, ethical judgment is considered an essential component of professionalism. This paper will point out the significance of teaching ethics, especially for the future AI professionals. It argues that education in Ethics should be incorporated in computing curricula. Experience from the course "Professional Ethics in Science and Engineering" given at Mälardalen University in Sweden is presented as an illustration.

1 Introduction

Computers play an essential role in today's industry, commerce, government, research, education, medicine, communication systems, entertainment, art and in many other fields. Professionals who contribute to the research, design, development, analysis, specification, certification, maintenance and evaluation of the many different applications of computer systems have a significant social impact.

Computing does not only produce artifacts, it also essentially changes our relation to world and to each other. To ensure that their efforts will be used for the general good, computing professionals must commit themselves to making computing a beneficial and respected profession, promoting an ethical approach to their professional practice.

Computing Curricula 2001, The Joint Task Force on Computing Curricula of IEEE Computer Society and Association for Computing Machinery (ACM), emphasizes strongly professional issues, as a *part of a core curriculum* for computing:
http://www.computer.org/education/cc2001/index.htm
Computing Curricula 2001.

The Engineering Criteria of the Accreditation Board for Engineering and Technology (ABET) (http://www.ele.uri.edu/People/Faculty/daly/criteria.2000.html Accreditation Board for Engineering and Technology (ABET) Engineering Criteria 2000 Third Edition), affirm that "Engineering programs must demonstrate that their graduates have an understanding of professional and ethical responsibility."

In spite of the above clear policy statements, professionalism and ethics are seldom present in undergraduate and graduate curricula in engineering and science. For example, in Sweden, only certain colleges and universities offer their students an opportunity to study professional ethics. Examples are The Royal Institute of Technology which gives courses in Engineering Ethics and Mälardalen University at which there is a course in Professional Ethics in Science and Engineering, presented for the first time in 2003. Beginning in September 2005, the Swedish Linköping University, the Norwegian University of Science and Technology NTNU, and Utrecht University jointly offer an Erasmus Mundus Master's programme in Applied Ethics (MAE), the courses which it offers including Computing Ethics.

The aim of this paper is to show that ethics is required for the successful performance of a computing professional. There are many ethical concerns which are characteristic for computing and Intelligent Systems as its subfield. The following is the list of questions to be addressed (Barger, 2001):

- Social context of computing
- Methods and tools of ethical argument

- Professional and ethical responsibilities
- Risks and liabilities of safety-critical systems
- Intellectual property
- Privacy and civil liberties
- Social implications of the Internet
- Computer crime
- Philosophical foundations of ethics

Engineering and research decisions are based on both engineering and ethical principles. We argue that training and education in professionalism and ethics should be an obligatory part of a computing professional's curriculum. Following this assumption we have developed a course "Professional Ethics in Science and Engineering" now included in the Computer Science, Software Engineering and Interactive Computing Curriculum at Mälardalen University (MdH), (Dodig-Crnkovic 2005). We give a short overview of this course, our experience and that of our students, and we present some direct consequences in an industrial context evident through the work of our industrial PhD students.

The rest of the paper is organized as follows. Section 2 discusses engineering, its impact and possible (unwanted) consequences of engineering decisions. Section 3 gives an introduction to Professional and Computer Ethics, the basic principles of Ethics and the specifics of Computer Ethics. Section 4 argues why Ethics should be taught to students of computing. Section 5 presents the course developed at Mälardalen University with the experiences of both the teachers and students. Finally, Section 6 summarizes the conclusions.

2 Engineering as a Large Scale Social Experimentation

"All products of technology present some potential dangers, and thus engineering is an inherently risky activity. In order to underscore this fact and help in exploring its ethical implications, we suggest that engineering should be viewed as an experimental process. It is not, of course, an experiment conducted solely in a laboratory under controlled conditions. Rather, it is an experiment on a social scale involving human subjects."

(Martin and Schinzinger, 1996)

There are uncertainties in every design process which are the result of our limited (finite) resources. Thus pharmaceuticals are tested in a limited (but of course representative) context. Construction materials are tested under certain conditions. Computer programs are tested for a large but finite number of cases. Intelligent systems behavior is likewise understood within some given limits. This implies that an engineered product may, sooner or later in its application, be used under conditions for which it has never been tested. New uncontrolled, unpredicted circumstances can appear. We expect the product to function properly, or at least safely, even in such circumstances. It is an engineer's responsibility to foresee and prevent as far as possible any severe consequences of product/system malfunction.

Modern history provides a wealth of examples of engineering failures with severe consequences. The intense media coverage of disasters such as the explosion of the Ariane V rocket in 1996, because of the incorrect reuse of some of the software developed for the Ariane IV and the radiation overdoses in the Therac-25 computerized linear accelerator for cancer treatment has increased the interest in engineering ethics. Major technical disasters are extremely costly but fortunately happen rarely. The judgment made by an engineer about what "rest (unknown, unaddressed) risk" in a safety analysis is acceptable, is to a high degree, an *ethical* one.

3 Computing Ethics

Computing Ethics might be defined as the analysis of the nature and social impact of computing technology and the corresponding formulation and justification of policies for the ethical use of such technology, (Moor, 1985).

Ethical problems arise most often when there are differences of judgment or expectations about what constitutes the true state of affairs or a proper course of action. An individual makes ethical decisions, in his/her capacity as a member of different groups. In order to make ethical decisions, an engineer or researcher interacts in many directions and within variety of contexts, each of which can show the actual situation in a different light. For example, solving the problem of the relation individual – colleagues – management could lead to certain choices, which e.g. do not necessarily coincide with the views of his or her own family or friends.

When faced with a moral/ethical dilemma, a professional must be able to make rational and well-motivated decisions. Courses in Ethics can help professionals by offering tools and methods useful in such situations.

3.1 Problems of Intelligent Systems Ethics as a Part of Computing Ethics

The basic principles of ethics are constant, no matter in which area they might be applied. The principles of Medical Ethics, Legal Ethics, and Computing Ethics are basically the same. However, only the first two of these are generally acknowledged as branches of Applied Ethics. In much the same way as for other Applied Ethics fields, new circumstances related to the computer do raise new questions about how general principles are to be applied, which results in policy vacuums designated thus by (Moor 1985) and further discussed in (Barger 2001; Tavani 2002 and Johnson 2003).

We argue that the social importance of the computer as a revolutionary machine together with its specific features give rise to new ethical problems and demands the introduction of the field of Computer Ethics. Some of the characteristic ethical problems of computing technology and in particular Intelligent Systems are listed in the following.

Logical malleability. Computers are malleable in a logical sense in that the results they produce can be shaped and molded to simulate any activity that can be characterized in terms of inputs, outputs, and associated logical operations (Moor, 1985). This is especially true of the AI field. Computers are used as tools for representation, modeling and simulation and they thereby have become a materialization of our conceptual knowledge of the world. For our epoch, they are The Revolutionary Machine in the same sense as the steam engine was for the industrial era. The ethical consequences of the fact that the computer is an artifact defining our contemporary culture are many. To this class of ethical issues belongs *Roboethics*[1], good examples are Stanford and Genova groups http://roboethics.stanford.edu/ and http://www.scuoladirobotica.it/roboethics/.

Speed. Speed and the simplicity of handling large amounts of data are connected with risks for unintentional transfer of incorrect data, as well as other ethical problems such as privacy and security intrusion because of unintended or uncontrolled movement of data.

Storage of huge amounts of data. When recorded and shared with other computers, information about people can be used to invade personal privacy and integrity in a way never before possible in history. The ease with which data saved in a computer can be manipulated, “as if they are greased” (Moor 2004) makes the use of surveillance, monitoring and spyware methods a simple technical operation. (Dodig-Crnkovic and Horniak 2006)

Identity vagueness. It is possible with a computer to steal another person’s identity, forge a message, or send a message anonymously. There is an ongoing ethical debate about the pros and contras of anonymity, and under which condition anonymity can be acceptable in communication. An interesting related topic is AI-aided misrepresentation of oneself. (Dodig-Crnkovic and Horniak 2005)

Copying. Images, text and sound can be copied with a computer by means of a few clicks and the copy easily used out of context or without attribution to the author. This has resulted in the ongoing discussion about intellectual property.

Openness and availability. Computer networks make it easy for the user to acquire a virtually unlimited amount of information. Propaganda or other sorts of disinformation and misinformation might be difficult to handle by some users. Cyberstalking is an example mentioned in Tavani (2002). Even spam and other unwanted messaging is a consequence of the openness of the system and the availability of data such as e-mail addresses. Of ethical interest is also spreading of the use of AI in excessively violent computer games and simulations. (Dodig-Crnkovic and Larsson 2005)

Globalization. Computer communication does not stop at national boundaries. What is considered legal in one country might not be allowed in some other country.

Safety. Interesting ethical problem is responsibility for errors in expert systems, safety critical systems (such as intelligent transportation systems, intelligent medical equipment, and automated computerized security equipment). Ethical aspects of self-modifying systems deserve special attention.

Power mediation. Computing is still a predominantly well-educated-younger-male-dominated field. The computer is increasingly becoming such a basic tool that it is a problem for certain social groups to be denied equal access to it, especially in the e-government era. The related ethical questions include the political power, equity, fairness and justice.

Privacy. Computers are powerful tools for collecting data about people in order to determine their habits and patterns of behavior and they may be used for both legal and illegal surveillance. This can be used to enhance public security and safety, but also to invade the privacy and personal integrity of the citizen. Related field within Intelligent Systems is automated monitoring of conversations (phone, email, and web). This may be solved by technical and ethics research into the development of protocols and policies that effectively balance privacy rights with Internet security. Intelligent systems are often related to this sort of ethical concerns.

New ethical challenges within Roboethics include the use of robots, ubiquitous sensing systems, direct neural interfaces and invasive nano devices that actualize ethical issues of self, control, privacy and access.

[1] Stanford group definition: Roboethics builds upon bioethics and the more specific topic of neuroethics, the engineering disciplines of robotics, nanotechnology and user interface design, and the bioengineering domain of direct brain interfaces, also called, depending on the specific project focus, neural prostheses, neural implants, brain-computer interfaces, or brain computer communication interfaces.

Ethical consequences of the increased use of robotics that will cause extensive social and economic changes, must be given due attention in the ethical theory and debate. For example, the relationship between humankind and robots, and other artificial autonomous agents should be discussed.

If the prediction of Veruggio (CNR-Robotlab, Italy) comes true, "… the net will be not only a network of computers, but of robots, and it will have eyes, ears and hands, it will be itself a robot." this envisaged robot-net will possess ethical challenges never seen before.

3.2 Codes of Ethics

How can we work to ensure that computing technology including Intelligent systems not only respects but also advances human values? It is necessary to integrate computing technology and human values in such a way that the technology protects and advances rather than harms human values. Much of the ground work in doing this is performed with the help of codes of ethics.

Professional societies in science and engineering publish their ethical codes or guidelines. (See the List of Codes of Ethics), which presents a sampling of ethical codes from societies of professional engineers and scientists. Some differ widely in their content, because of their origins and their specific purposes, but the main topics and the general ethical standards they articulate are similar.

Professional codes of ethics should be understood as conventions between professionals (Luegenbiehl 1983; Martin and Schinzinger 1995). Having a code of ethics allows an engineer to object to pressure to produce substandard work not merely as an ordinary citizen but as a professional engineer (or doctor, or scientist, etc.) who can say "As a professional, I cannot ethically put business concerns ahead of professional ethics." (Davis, 1991)

Harris and Pritchard (1995) summarize Unger's analysis of the possible functions of a code of ethics: "First, it can serve as a collective recognition by members of a profession of its responsibilities. Second, it can help create an environment in which ethical behavior is the norm. Third, it can serve as a guide or reminder in specific situations. Fourth, the process of developing and modifying a code of ethics can be valuable for a profession. Fifth, a code can serve as an educational tool, providing a focal point for discussion in classes and professional meetings. Finally, a code can indicate to others that the profession is seriously concerned with responsible, professional conduct."

Codes of ethics and case studies are always closely related. Without guiding principles, case studies are difficult to evaluate and analyze; without context, codes of ethics are incomprehensible. The best way to use these codes is to apply them in a variety of situations and study the results. It is from the back and forth evaluation of the codes and relevant cases that well-reasoned moral judgments can be arrived at.

4 Why Study Professional Ethics?

"Would you tell me, please, which way I ought to go from here?"

"That's depends a good deal on where you want to get to."...

L Carroll, Alice in Wonderland, Chapter VI, 1865

What is the point in studying ethics of computing? What can be gained from taking an Ethics course?

A Professional Ethics course is aimed to increase the ability of concerned engineers, researchers and citizens, to first recognize and then responsibly confront moral issues raised by technological activity. The goal is to cultivate moral autonomy, i.e. the skill and habit of to think rationally about ethical issues in the professional activity, and to increase the ability to think critically about moral matters. For the role of Computer Ethics in the Computer Science Curriculum, see Bynum (2004), and Moor (1985).

We are studying Ethicsin order to:

- deal with the true nature of computing as a service to other human beings (Gotterbarn, 1991).
- convey a sense of professional responsibility not covered in other courses
- sensitize students to Computer Ethics issues
- provide tools and methods for analyzing cases
- provide practice in applying the tools and methods to actual or realistic cases
- develop in the student good judgment and helpful intuitions - ethical autonomy.

These important topics are not addressed outside the computing curricula education.

5 Professional Ethics Course at MdH Sweden

Following the lines of reasoning presented in this article, we have developed a course in Professional Ethics at Mälardalen University, intended in the first place for Computer Science and Engineering students. The emphasis is on cultivating sensibility to ethical problems, the development of moral autonomy, ethical pluralism and critical thinking.

The course gives an insight into the ethical problems important for professionals in Engineering and Science. It forms a framework in which professional and ethical issues can be analyzed, and builds up an awareness of various views of ethical issues and the ethical responsibilities of professionals.

The course is delivered as a combination of lectures, guest lectures, classroom training (discussions), and training in writing essays. For Professional Ethics in Science and Engineering Course, see: http://www.idt.mdh.se/kurser/cd5590/

Our experiences of the course up to now (2003-2006) have been very positive. Students have participated actively in discussions, case studies and research on chosen topics. Even predominantly technically-minded students were able to assimilate and use philosophical concepts. The examination forms for the course were the writing of a research paper on an ethical topic of interest and an oral presentation of a chosen topic (such as safety and security, intellectual property, environmental ethics, privacy and personal integrity etc.) followed by an in-class discussion led by the students responsible for the actual presentation. Course evaluation results show that students experienced the course as very useful and relevant to their future professional activities.

Moreover, two industrial PhD students have included specific chapters on ethical aspects of their work in their PhD respective Licentiate Theses as a consequence of taking part in the Ethics course (Larsson M. 2004 and Larsson S. 2005). They have investigated ethical consequences of software testing practices, and the software development teams effectiveness related to different ethical attitudes. Three other students have published articles on their field of interest in international journals and at CEPE and E-CAP conferences.

6 Conclusions

The growing importance of computers in the society makes the study of Computing Ethics essential when it comes to issues such as safety, security, privacy, environmental impact and quality related to the research, design and development of computational systems. Of all fields within computing, Intelligent Systems have most interesting and diverse ethical aspects. It is therefore of special significance for the field to develop the ethical attitude. Good examples are Stanford and Genova Roboethics groups:
(http://roboethics.stanford.edu/
http://www.scuoladirobotica.it/roboethics/.

Ethics courses in science and engineering are aimed to increase the ability of future professionals to recognize and solve ethical problems, to accept different ethical perspectives and to adopt ethical pluralism. They develop the skill and habit of thinking rationally about ethical issues and in that way prepare students for the challenges of their profession. Experiences from the Professional Ethics in Science and Engineering Course at MdH are encouraging.

References

Barger R. N., Is Computer Ethics unique in relation to other fields of Ethics? , 2001 http://www.nd.edu/~rbarger/ce-unique.html

Bynum T. W., Computer Ethics in the Computer Science Curriculum. Supplementary materials for the book Computer Ethics and Professional Responsibility, edited by Terrell Ward Bynum and Simon Rogerson. 2004, http://www.southernct.edu/organizations/rccs/resources/teaching/teaching_mono/bynum/bynum_human_values.html

Davis, M., Thinking like an engineer: The place of a Code of Ethics in the practice of a profession", Philosophy and Public Affairs, 20.2 150, 1991

Dodig-Crnkovic G., On the Importance of Teaching Professional Ethics to Computer Science Students, Computing and Philosophy Conference, E-CAP 2004, Pavia, Italy; In: L. Magnani, Computing and Philosophy, Associated International Academic Publishers, 2005

Dodig-Crnkovic G., Larsson T., Game Ethics - Homo Ludens as a Computer Game Designer and Consumer. A special issue of IRIE on e-games, 2005

Dodig-Crnkovic G., Horniak V., Ethics and Ethics and Privacy of Communications in the e-Polis, Encyclopedia of Digital Government, 2006

Dodig-Crnkovic G., Horniak V., Good to Have Someone Watching Us from a Distance? Privacy vs. Security at the Workplace, Ethics of New Information Technology, Proceedings of the Sixth International Conference of Computer Ethics: Philosophical Enquiry, CEPE, University of Twente, The Netherlands; Brey P, Grodzinsky F and Introna L, Eds., 2005

Gotterbarn, D. The capstone course in Computer Ethics. Proceedings of the National Conference on Computing and Values, New Haven, 1991

Harris, C. E., Jr., Pritchard M. S. and Rabins M. J.,.Engineering Ethics: Concepts and Cases, Wadsworth Publishing, 1995

Johnson, D. G., Computer Ethics, in The Blackwell Guide to the Philosophy of Computing and Information (Blackwell Philosophy Guides), 65-75, 2003

Ladd, J., The quest for a Code of Professional Ethics: an intellectual and moral confusion, Ethical Issues in Engineering, Ed. Deborah G. Johnson. Englewood Cliffs, NJ: Prentice-Hall, 130-136. , 1991

Larsson M., Predicting Quality Attributes in Component-based Software Systems, PhD Thesis, Mälardalen University Press, Sweden, ISBN: 91-88834-33-6, 2004

Larsson S., Improving Software Product Integration, Licenciate Thesis, Mälardalen University Press, Sweden, ISBN 91-88834-65-4, 2005

Luegenbiehl, H. C., Codes of Ethics and the moral education of engineers, Business and Professional Ethics Journal 2 (1983): 41-61. Rpt. in Ethical Issues in Engineering . Ed. D. G. Johnson. Englewood Cliffs, NJ: Prentice-Hall, 137-154., 1991

Martin, M.W., Schinzinger, R., Ethics in Engineering, McGraw-Hill, 1996

Moor J., What is Computer Ethics, Metaphilosophy 16(4): 266-75., 1985

Moor J., 2004, Towards a Theory of Privacy in the Information Age. Computer Ethics and Professional Responsibility. Edited by Terrell W Bynum and S Rogerson. Blackwell Publishing

Tavani H., 2002, The uniqueness debate in Computer Ethics: What exactly is at issue, and why does it matter?, Ethics and Information Technology 4: 37–54

Codes of Ethics Links

IEEE Code of Ethics
www.ieee.org/about/whatis/code.html

ACM Code of Ethics and Professional Conduct
http://www.acm.org/constitution/code.html

Responsible Conduct In Research
http://www.nap.edu/readingroom/books/obas

Codes of Ethics for Engineering
http://www.iit.edu/departments/csep/PublicWWW/codes/engineer.html

Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering, A Volume of the Computing Curricula Series SE2004
http://sites.computer.org/ccse/

American Society of Civil Engineers Code of Ethics
http://www.asce.org/inside/codeofethics.cfm

Software Engineering Code Of Ethics And Professional Practice
http://www.computer.org/tab/seprof/code.htm

Ethics in Computing "site map"
http://legacy.eos.ncsu.edu/eos/info/computer_ethics/

Codes of Ethics Online
http://www.iit.edu/departments/csep/PublicWWW/codes/

Analysis of Interdisciplinary Text Corpora

Matti Pöllä[⋆]
Timo Honkela[⋆]
[⋆]Helsinki University of Technology
Adaptive Informatics Research Centre
P.O. Box 5400 FI-02015 TKK, Finland
{matti.polla, timo.honkela}@tkk.fi

Henrik Bruun[†]
[†]University of Turku
Institutions and Social Mechanisms IASM
FI-20014 Turku, Finland
henbru@utu.fi

Ann Russell[‡]
[‡]University of Toronto
Knowledge Translation Program at St. Michael's Hospital
russell@smh.toronto.on.ca

Abstract

Computational analysis of natural language is often focused on the syntactic structure of language–often with no regard to the overall context and area of expertise of the analyzed text. In this paper, we present a means of analyzing text documents from various areas of expertise to discover groups of thematically similar texts with no prior information about the topics. The presented results show how a relatively simple keyword analysis combined with a SOM projection can be very descriptive in terms of analyzing the contextual relationships between documents and their authors.

1 Introduction

When reading a text document, humans process information not only by parsing individual sentences and their meanings. Instead, a large collection of prior information about the subject at hand is often necessary–especially in the communication between content experts in highly specialized professions. However, computational processing of natural language is often limited to processing information sentence-by-sentence. While this approach can be sufficient for example in elementary machine translation tasks, a more abstract view about the overall nature of the document would greatly facilitate automated processing of natural language in many applications, such as machine translation and information retrieval. Eventually, this kind of information about the document can be used further for interdisciplinary (within language) translation tasks.

In this paper, a method based on automatically extracted keywords combined with the Self-Organizing Map Kohonen (1995) is presented for text document analysis to visualize the relationships of the documents and their authors. The significance of computational tools for interdisciplinary language analysis has been outlined in Bruun and Laine (2006) referring to previous success in the analysis of text document using the WEBSOM method Honkela et al. (1997).

In section 2 we describe the applied method for keyword selection and in section 3 how the SOM is employed for analysis. In section 4 we present results of an experiment that used corpora from two distinctive areas of expertise. In section 5 some conclusions are made about the applications of the presented method.

2 Selecting descriptive keywords

Our analysis of text documents attempts to extract information about the area of expertise of the document using a set of keywords which are extracted from the documents automatically. To extract relevant keywords for each text document the frequency of each word is examined as an indicator of relevance. As the word frequency distribution for any source of natural language has a Zipf'ian form, special care has to be taken to filter out words, which occur frequently in all documents but are irrelevant in describing the topic of the document (such as 'the', 'a', 'an', 'is' etc.). Figure 1 shows a typical word frequency distribution for a text document in which only a small class of words have a large frequency with the rest of the words having an exponential-like distribution. In Figure 2 the word-use for text documents is visualized in a matrix form. In this figure, each row corresponds to a text document and each column to a specific word with

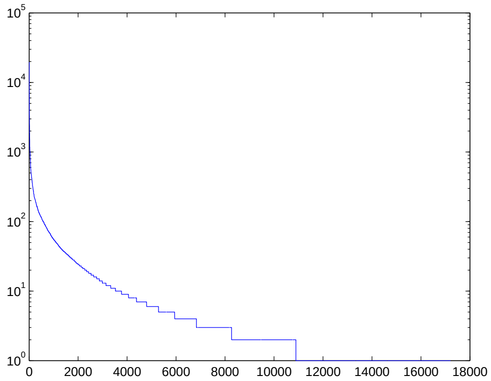


Figure 1: The word-use frequency for natural language resulting in a Zipf'ian distribution.

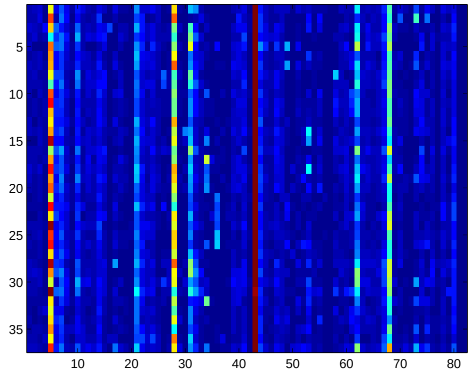


Figure 2: A visualization of the word frequency: each row corresponds to a text document and each column the frequency of a certain word. Note how some of the most common words (which appear in every document) form a light vertical stripe. Other more discipline-specific words can be seen as disconnected light dots.

a light color shade indicating frequently used words. The interesting document-specific frequent words can be seen as individual spots whereas the frequent but irrelevant words can be seen as vertical light stripes.

To distinguish between words that appear frequently in a specific document from the words that are frequent in language in general, a straightforward comparison can be made by relating the frequency f_w^D of the word w in the examined document D and the corresponding frequency f_w^{ref} in another corpus that can be considered more neutral in terms of using less discipline-specific vocabulary Damerau (1993).

However, by comparing the word frequencies directly we would need to normalize the frequencies for each document to get comparable results. For this reason we use the rank of occurrence of each word instead of the frequency itself. Using this method we can compute a relevance score for each word w_i

$$\text{score}(w_i) = \frac{\text{rank}_D(w_i)}{\text{rank}_{\text{ref}}(w_i)} \tag{1}$$

where $\text{rank}_D(w_i)$ is the rank of word w_i in the list of most common words in document D and $\text{rank}_{\text{ref}}(w_i)$ is the corresponding rank in the reference corpus.

3 SOM analysis of keyword occurrence in documents

After applying the keyword extraction method to each document we have gathered an overall set of keywords. We can use this result in a further analysis where the objective is to find clusters of similar documents and analyze the relationships between the documents and their authors by using the Self-Organizing Map algorithm Kohonen (1995).

The SOM algorithm is a widely accepted tool for creating a two-dimensional visual map representation of a high dimensional data set. In this case we use the occurrence of the extracted keywords as a vector representation of the text documents. This way each document D_i will be represented by an n-dimensional vector

$$D = [f_{w_1}, f_{w_2}, f_{w_3}, ..., f_{w_n}]$$

in which $f_{w_j}^d$ is the frequency of keyword j in the document. For simplicity, we used a binary value merely to indicate whether the keyword is present in the document or not.

As a result, documents with similar keyword occurrence profiles will locate close to each other in the SOM projection. Correspondingly documents that do not share similar keyword occurrence profiles will be located further away from each other.

4 Experiments

To test the proposed method, we selected two sets of documents from two distinctive fields of expertise: the first corpus A was collected from scientific articles published in the proceedings of the AKRR'05 conference with the topics of the papers mainly in the fields of computer science, cognitive science, language technology and bioinformatics Honkela et al. (2005); Bounsaythip et al. (2005); Russell et al. (2005). Corpus B consists of a collection of articles published by the Laboratory of Environmental Protection at Helsinki University of Technology with the topics ranging from social sciences to environmental managing. Henceforth these will be referred to as corpus A and corpus B.

Table 1: The two corpora used in the experiments

	Corpus A	Corpus B	Reference
Words	190 107	372 836	92 360 430
Files	45	37	308

At first, some preprocessing was done to the text material. This included removing all punctuation symbols and transforming all symbols to lowercase letters. This was done because the downside of adding noise caused by the loss of information due to removing punctuation was considered smaller than the benefits of preprocessing.

For the keyword extraction phase, the Europarl corpus Koehn (a,b) was selected as a reasonably discipline-neutral reference material. In Figure 3 a sample of the extracted keyword set is shown. The selection method has resulted in a fairly good set of keywords which describe the contents of the documents without being too general.

```
architecture bayesian best better
biological clustering computer denmark
distance distributed effect experiments
expression extraction forestry found
four further growth impact
interdisciplinary interviewees
introduction kola learned makes meaning
mediated morphs necessary objects often
organized panel panels paradigm
parameters part personal points present
press product publications
recommendations rho science
selforganizing shows significance
subject test thought together
understand visualization weights words
xml
```

Figure 3: A sample of the keyword set extracted from corpus A and B.

Finally, the result of the SOM projection of the keyword frequency data is seen in Figure 4. In this figure, the documents of corpora A and B are shown.

Looking at the distribution of the articles in the U-matrix projection in Figure 4, we can see a clear cluster of papers in corpus A in the top of the map. Most of the map is occupied by the mid-upper part of documents from corpus B. Clusters inside these areas can also bee seen indicating groups of thematically similar documents inside the corpus.

Another result of the experiment is visualized in Figure 5 where the individual component planes of the SOM projection are shown according to several keywords. In this figure a light gray shade corresponds to an area where the keyword in question has been used and a dark area to the areas where it does not occur. By combining this information with the locations of the individual documents in Figure 4 we can see what kind of topics are covered in the documents.

Some of the keywords are common in most of the documents and do not discriminate corpora A and B significantly. For example, the usage of the words 'learning', 'language', 'theory', 'knowledge', and 'complex' is distributed all across the map and are thus used commonly in nearly all documents. This observation is well aligned with our intuition about the overall content of the documents.

By contrast, some of the keywords appear only in a small group of documents are less general. For example the words 'interdisciplinarity', 'ecology', and 'sociobiology' appear almost exclusively in the lower part of the map corresponding to corpus B. Other terms such as 'bayesian', 'algorithm', and 'selforganizing' can be seen mostly in the area occupied by the documents of corpus A.

In addition to grouping documents from different sources, we can see sub-clusters for example in the bottom left corner of the map. In this small area the words 'reindeer', 'herders', and 'forest' occur in the documents B_{11} B_{29} and B_{42} which form a subgroup inside corpus B.

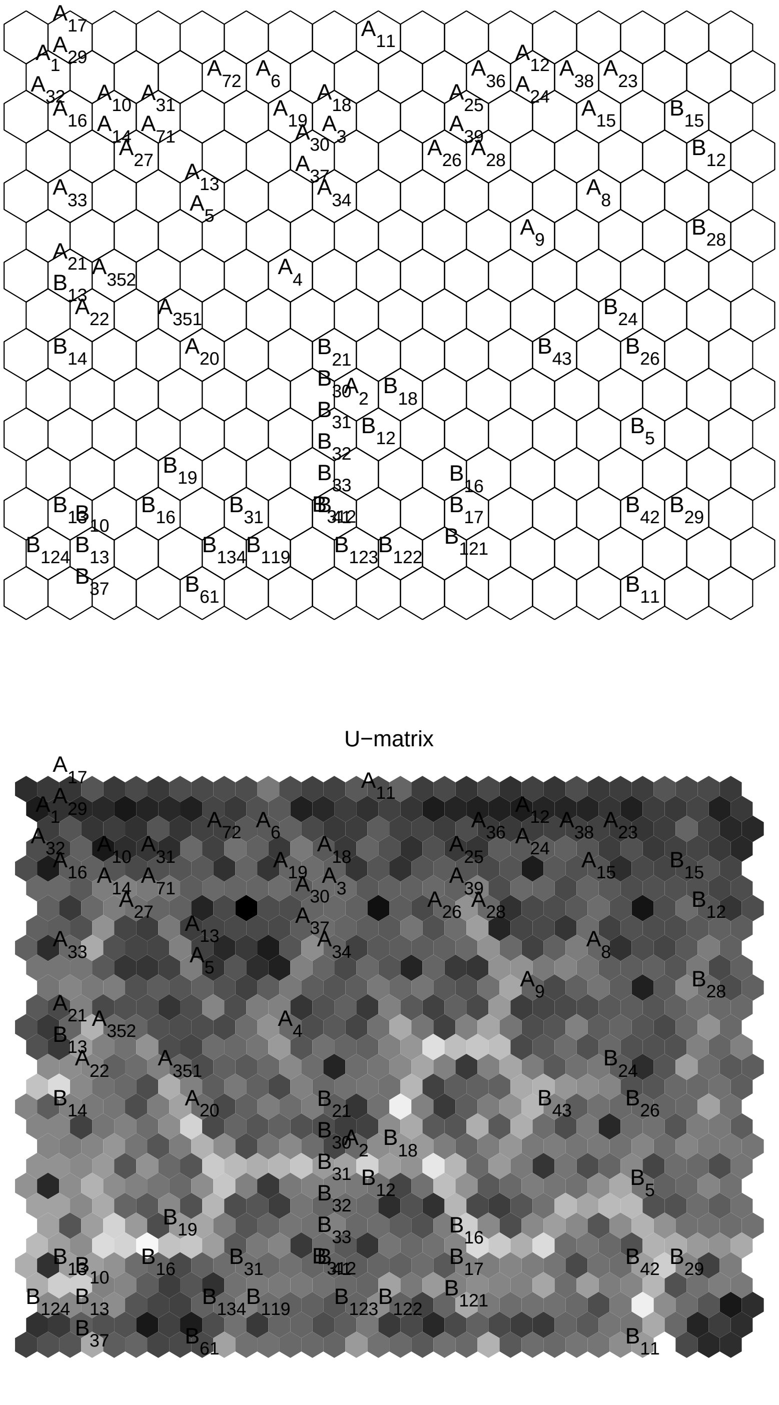


Figure 4: The documents of corpora A and B in a SOM projection (top) and a U-matrix visualization of the document map (bottom). The documents of corpus A have formed a cluster in the upper part of the figure while the lower half of the map is occupied with documents from corpus B. Clusters within the area dominated by documents from corpus B can also be seen (e.g., B_{11} B_{29} and B_{42}).

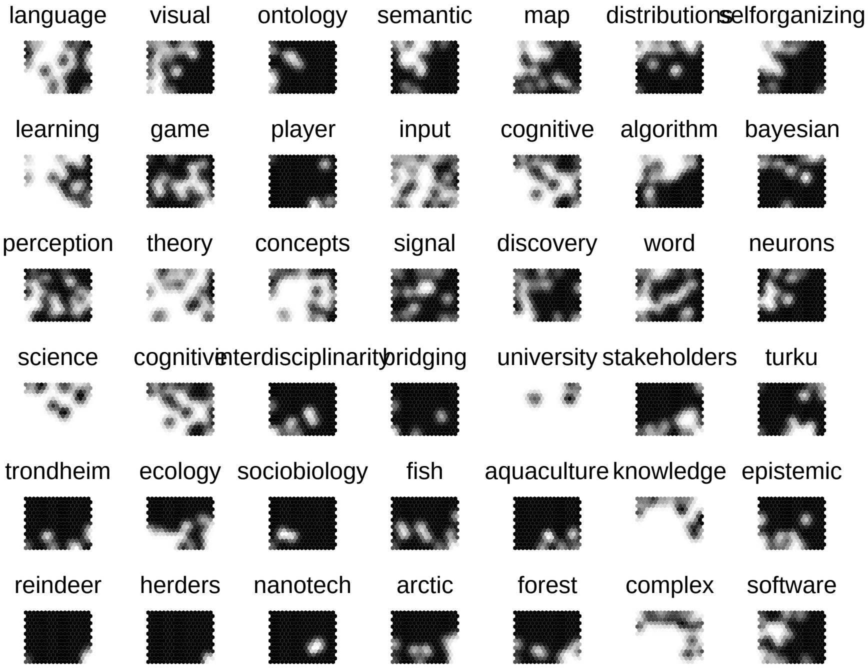


Figure 5: Component plane visualizations of 42 keywords used in the analysis. In the figures, light shades correspond to occurrence of the keyword in the specific region of the map.

5 Discussion

In this paper, a novel method was presented for content analysis of text documents and their authors using automatically extracted keyword selection combined with a Self-Organizing Map analysis.

Our results show that the adopted word frequency analysis can be effective in describing the contents of a document. A SOM analysis of the keyword occurrence was successful in clustering not only documents from different sources but also similar documents in the same corpus.

However, our current efforts have been restricted to isolated single words ignoring the potential of analyzing variable-length phrases. An analysis of keyphrases consisting of several words would evidently be beneficial, since the topics of scientific articles often consist of two-word concepts such as 'neural network' or 'signal processing'.

Despite the limitations of the current implementation, our experiments have shown that a combination of an automatic keyword extraction scheme combined with a clustering algorithm can be effective in describing the mutual similarities of text documents. Our statistical approach to the task has the additional benefit of being independent of the language that is being processed as no prior information about the processed language syntax is encoded into the algorithm.

The possible application areas of the proposed method are diverse. For example, in machine translation the acquired prior information about the context of the information could be used to disambiguate words or acronyms that have different meanings depending on the context. In information retrieval the method could be used to help in navigating through document collections as the search engine could point out keywords that are considered relevant but are omitted in the search. Finally, automated methods that determine similarities and dissimilarities across diverse and vast knowledge bases have the potential to advance scientific discoveries both within and across disciplines.

References

Catherine Bounsaythip, Jaakko Hollmén, Samuel Kaski, and Matej Oreĺic, editors. *Proceedings of KRBIO'05, International Symposium of the Knowledge Representation in Bioinformatics*. Helsinki University of Technology, Espoo, Finland, June 2005.

Henrik Bruun and Sampsa Laine. Using the self-organizing map for measuring interdisciplinary research. Manuscript, 2006.

Fred Damerau. Generating and evaluating domain-oriented multi-word terms from texts. *Inf. Process. Manage.*, 29(4):433–448, 1993.

Timo Honkela, Samuel Kaski, Krista Lagus, and Teuvo Kohonen. WEBSOM—self-organizing maps of document collections. In *Proceedings of WSOM'97, Workshop on Self-Organizing Maps, Espoo, Finland, June 4-6*, pages 310–315. Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland, 1997.

Timo Honkela, Ville Könönen, Matti Pöllä, and Olli Simula, editors. *Proceedings of AKRR'05, International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning*. Helsinki University of Technology, Espoo, Finland, June 2005.

Philipp Koehn. Europarl parallel corpus, a. http://people.csail.mit.edu/koehn/publications/europarl/.

Philipp Koehn. Europarl: A multilingual corpus for evaluation of machine translation. Unpublished manuscript, b.

Teuvo Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, Berlin, Heidelberg, 1995. (Second Extended Edition 1997).

Ann Russell, Timo Honkela, Krista Lagus, and Matti Pöllä, editors. *Proceedings of AMKLC'05, International Symposium on Adaptive Models of Knowledge, Language and Cognition*. Helsinki University of Technology, Espoo, Finland, June 2005.

Reducing High-Dimensional Data by Principal Component Analysis vs. Random Projection for Nearest Neighbor Classification

Sampath Deegalla[⋆]
[⋆]Dept. of Computer and Systems Sciences,
Stockholm University and Royal Institute of Technology,
Forum 100, SE-164 40 Kista, Sweden.
si-sap@dsv.su.se

Henrik Boström[†]
[†]Dept. of Computer and Systems Sciences,
Stockholm University and Royal Institute of Technology,
Forum 100, SE-164 40 Kista, Sweden.
henke@dsv.su.se

Abstract

The computational cost of using nearest neighbor classification often prevents the method from being applied in practice when dealing with high-dimensional data, such as images and micro arrays. One possible solution to this problem is to reduce the dimensionality of the data, ideally without loosing predictive performance. Two different dimensionality reduction methods, principal component analysis (PCA) and random projection (RP), are compared w.r.t. the performance of the resulting nearest neighbor classifier on five image data sets and two micro array data sets. The experimental results show that PCA results in higher accuracy than RP for all the data sets used in this study. However, it is also observed that RP generally outperforms PCA for higher numbers of dimensions. This leads to the conclusion that PCA is more suitable in time-critical cases (i.e., when distance calculations involving only a few dimensions can be afforded), while RP can be more suitable when less severe dimensionality reduction is required. In 6 respectively 4 cases out of 7, the use of PCA and RP even outperform using the non-reduced feature set, hence not only resulting in more efficient, but also more effective, nearest neighbor classification.

1 Introduction

With the development of technology, large volumes of high-dimensional data become rapidly available and easily accessible for the data mining community. Such data includes high resolution images, text documents, and gene expressions data and so on. However, high dimensional data puts demands on the learning algorithm both in terms of efficiency and effectiveness. The *curse of dimensionality* is a well known phenomenon that occurs when the generation of a predictive model is mislead by an overwhelming number of features to choose between, e.g., when deciding what feature to use in a node of a decision tree (Witten and Frank, 2005). Some learning methods are less sensitive to this problem since they do not rely on choosing a subset of the features, but instead base the classification on all available features. Nearest neighbor classifiers belong to this category of methods (Witten and Frank, 2005). However, although increasing the number of dimensions does not typically have a detrimental effect on predictive performance, the computational cost may be prohibitively large, effectively preventing the method from being used in many cases with high-dimensional data.

In this work, we consider two methods for dimensionality reduction, principal component analysis (PCA) and random projection (RP) (Bingham and Mannila, 2001; Fradkin and Madigan, 2003; Fern and Brodley, 2003; Kaski, 1998). We investigate which of these is most suited for being used in conjunction with nearest neighbor classification when dealing with two types of high-dimensional data: images and micro arrays.

In the next section, we provide a brief description of PCA and RP and compare them w.r.t. computational complexity. In section three, we present results from a comparison of the methods when used together with nearest neighbor classification on five image data sets and two micro array data sets, together with an analysis of the results. In section four, we discuss some related work, and finally, in section five, we give some concluding remarks and point out some directions for future work.

2 Dimensionality reduction methods

Principal component analysis (PCA) and Random projection (RP) are two dimensionality reduction methods that have been used successfully in conjunction with learning methods (Bingham and Mannila, 2001; Fradkin and Madigan, 2003). PCA is the most well-known and popular of the above two, whereas RP is more recently gaining popularity among researchers (Bingham and Mannila, 2001; Fradkin and Madigan, 2003; Fern and Brodley, 2003; Kaski, 1998), not least by being much more efficient.

Principal component analysis (PCA)

PCA is a technique which uses a linear transformation to form a simplified data set retaining the characteristics of the original data set.

Assume that the original data matrix contains d dimensions and n observations and it is required to reduce the dimensionality into a k dimensional subspace. This transformation is given by

$$Y = E^T X \tag{1}$$

Here $E_{d\times k}$ is the projection matrix which contains k eigen vectors corresponds to the k highest eigen values, and where $X_{d\times n}$ is a mean centered data matrix. We have followed the Singular Value Decomposition (SVD) of the data matrix to calculate principal components.

Random projection(RP)

Random projection is based on matrix manipulation, which uses a random matrix to project the original data set into a low dimensional subspace (Bingham and Mannila, 2001; Fradkin and Madigan, 2003).

Assume that it is required to reduce the d dimensional data set into a k dimensional set where the number of instances are n,

$$Y = RX \tag{2}$$

Here $R_{k\times d}$ is the random matrix and $X_{d\times n}$ is the original matrix. The idea for random projection originates from the Johnson-Lindenstrauss lemma (JL) (Dasgupta and Gupta, 1999). It states that a set of n points could be projected from $R^d \rightarrow R^k$ while approximately preserving the Euclidean distance between the points within an arbitrarily small factor. For the theoretical effectiveness of random projection method, see (Fradkin and Madigan, 2003).

Several algorithms have been proposed to generate the matrix R according to the JL, and Achlioptas's way of constructing R have received much attention in the literature (Bingham and Mannila, 2001; Fradkin and Madigan, 2003). According to Achlioptas (Achlioptas, 2001), the elements of the random matrix R can be constructed in the following way:

$$r_{ij} = \begin{cases} +\sqrt{3} & \text{with} \quad P_r = \frac{1}{6}; \\ 0 & \text{with} \quad P_r = \frac{2}{3}; \\ -\sqrt{3} & \text{with} \quad P_r = \frac{1}{6}. \end{cases} \tag{3}$$

An analysis of the computational complexity of random projection shows that it is very efficient compared to principal component analysis. Random projection requires only $O(dkn)$ whereas principal component analysis needs $O(d^2n)+O(d^3)$ (Bingham and Mannila, 2001). If rank of matrix X is r then computational complexity for SVD is $O(drn)$ (Bingham and Mannila, 2001).

3 Empirical study

3.1 Experimental setup

In comparison of the use of PCA and RP for nearest neighbor classification, five image data sets (IRMA (Lehmann et al., 2000),COIL-100 (Nene et al., 1996), ZuBuD (H.Shao et al., 2003),MIAS (MIAS data set), Outex (Ojala et al., 2002)) and two micro array data sets (Colon Tumor (Alon et al., 1999), Leukemia (Golub et al., 1999)) have been used. The image data sets consist of two medical image data sets (IRMA,MIAS), two object recognition data sets (COIL-100,ZuBuD) and a texture analysis data set (Outex - *TC_00013*). The IRMA (Image Retrieval and Medical Application) data set contains radiography images of 57 classes, where the quality of the images varies significantly. The COIL-100 (Columbia university image library) data set consists of images of 100 objects, while ZuBuD (Zurich Building Image Database) contains images of 201 buildings in Zurich city. MIAS (The Mammography Image Analysis Society) mini mammography database contains mam-

Table 1: Description of data

Data set	Instances	Attributes	# of Classes
IRMA	9000	1024	57
COIL100	7200	1024	100
ZuBuD	1005	1024	201
MIAS	322	1024	7
Outex	680	1024	68
Colon Tumor	62	2000	2
Leukemia	38	7129	2

mography images of 7 categories and finally Outex (University of Oulu Texture Database) image data set contains images of 68 general textures. Two micro array data sets are also included in the study: Leukemia (ALL-AML [1]) (Golub et al., 1999) data set and Colon Tumor (Alon et al., 1999).

For all image data sets, colour images have been converted into gray scale images with 256 gray values and then resized into 32×32 pixel sized images. The original matrices consist of pixel brightness values as features. Therefore, all image data sets contain 1024 attributes. The number of attributes for the micro array data sets and the number of instances for all data sets are shown in Table 1.

The Waikato Environment for Knowledge Analysis (WEKA) (Witten and Frank, 2005) implementation of the nearest neighbor classifier was used for this study. PCA was done using MATLAB whereas WEKA's RP filter implementation was used for RP. The accuracies were estimated using ten-fold cross-validation, and the results for RP is the average from 30 runs to account for its random nature.

3.2 Experimental results

The accuracies of using a nearest neighbor classifier on data reduced by PCA and RP, as well as without dimensionality reduction, are shown in Figure 1 for various number of resulting dimensions. Table 2 shows the highest accuracies obtained by using the two dimensionality reduction methods with the corresponding number of dimensions. Table 3 shows the accuracies obtained for PCA and Table 4 shows the accuracies and standard deviations (SD) for RP.

3.3 Analysis

The experimental results show that reducing the dimensionality by using PCA results in a higher accuracy than by using RP for all data sets used in this study. In Table 2, it can be seen that only a few principal components is required for achieving the highest accuracy. However, RP typically requires a larger number of dimensions compared to PCA to obtain a high accuracy.

Classification accuracy using PCA typically has its peak for a small number of dimensions, after which the accuracy degrades. In contrast to this, the accuracy of RP generally increases with the number of dimensions. For large numbers of dimensions, RP generally outperforms PCA.

It can also be seen that the time required for doing a prediction is reduced when using a dimensionality reduction method as shown in Table 5. Even though nearest neighbor classification is often less efficient when compared to other learning methods, the time required may be significantly reduced by using the dimensionality reduction while still maintaining a high accuracy.

The results can be interpreted as if one can only afford a few number of dimensions (e.g. due to time constraints during classification), PCA appears to be more suitable than RP. On the other hand, for large numbers of dimensions, RP appears more suitable.

It can be noted that in 6 respectively 4 cases out of 7, the use of PCA and RP, even outperform using the non-reduced feature set, hence demonstrating that dimensionality reduction may not only lead to more efficient, but also more effective, nearest neighbor classification.

4 Related work

Fradkin and Madigan (Fradkin and Madigan, 2003) have compared PCA and RP with decision trees (C4.5), k-nearest-neighbor classification with k=1 and k=5 and support vector machines for supervised learning. In their study, PCA outperforms RP, but it was also realized that there was a significant computational overhead of using PCA compared using RP.

Bingham and Mannila (Bingham and Mannila, 2001) have also compared RP with several other dimensionality reduction methods such as PCA, singular value decomposition (SVD), Latent semantic indexing (LSI) and Discrete cosine transform (DCT) for image and text data. The criteria chosen for the comparison was the amount of distortion caused on the original data and the computational complexity.

[1] ALL: acute lymphoblastic leukemia
AML: acute myeloid leukemia

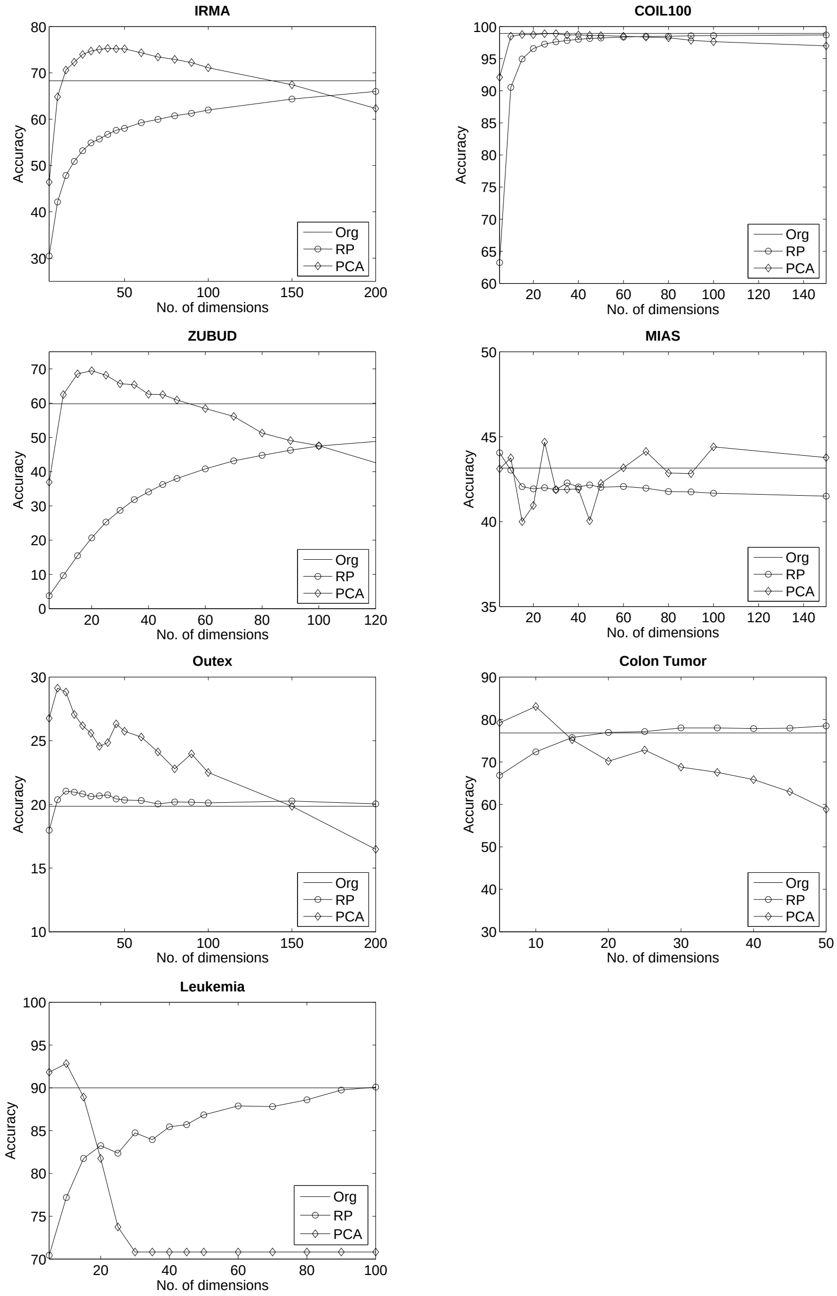


Figure 1: Comparison of the accuracies among Original and PCA and RP based attributes.

Table 2: The highest prediction accuracy (%) of PCA and RP with minimum number of dimensions

	IRMA		COIL-100		ZuBuD		MIAS		Outex		Colon		Leukemia	
PCA	**75.30**	(40)	98.90	(30)	**69.46**	(20)	**53.76**	(250)	**29.12**	(10)	**83.05**	(10)	**92.83**	(10)
RP	67.01	(250)	98.79	(250)	54.01	(250)	44.05	(5)	21.04	(15)	80.22	(150,200)	91.32	(150)
All	68.29	(1024)	**98.92**	(1024)	59.81	(1024)	43.15	(1024)	19.85	(1024)	76.83	(2000)	90.00	(7129)

They also extended their experiments to determine the effects on noisy images and noiseless images.

Fern and Brodley (Fern and Brodley, 2003) have used Random projections for unsupervised learning. They have experimented with using RP for clustering of high dimensional data using multiple random projections with ensemble methods. Furthermore, they also proposed a novel approach based on RP for clustering, which was compared with PCA.

Kaski (Kaski, 1998) used RP in the WEBSOM system for document clustering. RP was compared to PCA for reducing the dimensionality of the data in order to construct Self-Organized Maps.

5 Concluding remarks

We have compared the use of PCA and RP for reducing dimensionality of data to be used by a nearest neighbor classifier on five image data sets and two micro array data sets. It was observed that for all data sets, the use of PCA resulted in a higher accuracy compared to using RP. It was also observed that PCA is more effective for severe dimensionality reduction, while RP is more suitable when keeping a high number of dimensions (although a high number is not always optimal w.r.t. accuracy).

In 6 cases out of 7 for PCA, and in 4 cases out of 7 fo RP, a higher accuracy was obtained than using nearest-neighbor classification with the non-reduced feature set. This shows that dimensionality reduction may not only lead to more efficient, but also more effective, nearest neighbor classification.

Directions for future work include considering other types of high-dimensional data to gain a further understanding of the type of data for which each of the two dimensionality reduction techniques is best suited, as well as considering other dimensionality reduction methods for nearest neighbor classification.

Acknowledgements

The authors would like to thank T.M. Lehmann, Dept. of Medical Informatics, RWTH Aachen, Germany for providing the ”10000 IRMA images of 57 categories”.

Financial support for the first author by SIDA/SAREC is greatly acknowledged.

References

D. Achlioptas. Database-friendly random projections. In *ACM Symposium on the Principles of Database Systems*, pages 274–281, 2001.

U. Alon, N. Barkai, D. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. In *Proc. Natl. Acad. Sci. USA*, volume 96, pages 6745– 6750, 1999. Data set : http://sdmc.lit.org.sg/GEDatasets/Datasets.html.

E. Bingham and H. Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Knowledge Discovery and Data Mining*, pages 245–250, 2001.

S. Dasgupta and A. Gupta. An elementary proof of the Johnson-Lindenstrauss lemma. Technical Report TR-99-006, International Computer Science Institute, Berkeley, CA, 1999.

X. Z. Fern and C. E. Brodley. Random projection for high dimensional data clustering: A cluster ensemble approach. In *Proceedings of the Twentieth International Conference of Machine Learning*, 2003.

D. Fradkin and D. Madigan. Experiments with Random Projections for Machine Learning. In *KDD ’03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 517–522, 2003.

T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999. Data set : http://sdmc.lit.org.sg/GEDatasets/Datasets.html.

H.Shao, T. Svoboda, and L. Van Gool. Zubud - zurich building database for image based recognition. Technical report, Computer Vision Lab, Swiss Federal Institute of Technology, Switzerland, 2003. Data set : http://www.vision.ee.ethz.ch/showroom/zubud/-index.en.html.

S. Kaski. Dimensionality Reduction by Random Mapping: Fast Similarity Computation for Clustering. In *Proceedings of IJCNN'98, International Joint Conference on Neural Networks*, volume 1, pages 413–418, Piscataway, NJ, 1998. IEEE Service Center.

T. M. Lehmann, B. B. Wein, J. Dahmen, J. Bredno, F. Vogelsang, and M. Kohnen. Content-based image retrieval in medical applications: a novel multistep approach. In M. M. Yeung, B.-L. Yeo, and C. A. Bouman, editors, *Proceedings of SPIE: Storage and Retrieval for Media Databases 2000*, volume 3972, pages 312–320, 2000. Data set : http://phobos.imib.rwth-aachen.de/irma/datasets_en.php.

MIAS data set. MIAS data set : http://www.wiau.man.ac.uk/services/MIAS/-MIASmini.html.

S. A. Nene, S. K. Nayar, and H. Murase. Columbia Object Image Library: COIL-100. Technical report, CUCS-006-96, February 1996. Data set : http://www1.cs.columbia.edu/CAVE/research/-softlib/coil-100.html.

T. Ojala, T. Maenpaa, M. Pietikainen, J. Viertola, J. Kyllonen, and S. Huovinen. Outex - new framework for empirical evaluation of texture analysis algorithms. In *ICPR '02: Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 1*, 2002. Data set : http://www.outex.oulu.fi.

I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2005.

Table 3: Classification accuracies (%) on PCA

Dim. (k)	IRMA	COIL-100	ZuBuD	MIAS	Outex	Colon	Leukemia
5	46.41	92.10	36.92	43.11	26.76	79.21	91.83
10	64.83	98.50	62.50	43.75	29.12	83.05	92.83
15	70.62	98.75	68.56	40.00	28.82	75.21	88.92
20	72.31	98.74	69.46	40.95	27.06	70.17	81.75
25	73.97	98.88	68.16	44.68	26.18	72.81	73.75
30	74.71	98.90	65.68	41.88	25.59	68.76	70.83
35	75.06	98.68	65.38	41.90	24.56	67.55	70.83
40	75.30	98.71	62.60	41.92	24.85	65.83	70.83
45	75.18	98.63	62.50	40.05	26.32	62.98	70.83
50	75.18	98.58	60.91	42.25	25.74	58.86	70.83
60	74.33	98.49	58.43	43.17	25.29	37.31	70.83
70	73.44	98.36	56.13	44.13	24.12	41.76	70.83
80	72.91	98.26	51.27	42.86	22.79	41.76	70.83
90	72.22	97.86	49.07	42.83	23.97	41.76	70.83
100	71.11	97.65	47.58	44.40	22.50	41.76	70.83
150	67.43	97.00	35.14	43.77	19.85	41.76	70.83
200	62.32	96.00	27.57	40.67	16.47	41.76	70.83
250	57.50	94.65	21.01	53.76	12.79	41.76	70.83

Table 4: Classification accuracies (%) on RP

	IRMA		COIL-100		ZuBuD		MIAS		Outex		Colon		Leukemia	
Dim.(k)	Acc.	SD	Acc.	SD	Acc.	SD	Acc.	SD	Acc.	SD	Acc.	SD	Acc.	SD
5	30.46	2.77	63.24	2.69	3.77	0.86	44.05	2.65	17.97	2.01	66.83	8.39	70.44	11.10
10	42.14	3.62	90.54	0.71	9.64	1.66	43.03	2.50	20.37	1.39	72.37	5.06	77.19	7.70
15	47.86	3.93	94.95	0.50	15.46	1.63	42.60	2.41	21.04	1.51	75.75	5.69	81.75	5.52
20	50.89	3.99	96.57	0.35	20.66	1.70	41.93	1.81	20.96	1.25	76.94	5.32	83.25	7.64
25	53.19	3.75	97.26	0.28	25.28	1.33	42.00	2.00	20.82	1.33	77.15	5.82	82.37	6.08
30	54.89	3.57	97.62	0.28	28.69	1.53	41.88	2.27	20.62	1.20	78.01	4.97	84.74	5.01
35	55.71	3.73	97.83	0.24	31.83	1.54	42.28	2.38	20.67	1.27	78.01	4.38	83.95	5.98
40	56.72	3.62	98.01	0.21	34.08	1.55	42.04	2.35	20.75	1.16	77.85	4.82	85.44	4.98
45	57.61	3.18	98.14	0.19	36.23	1.67	42.16	2.40	20.43	0.77	77.96	4.09	85.70	5.21
50	58.03	2.93	98.22	0.17	37.99	1.52	42.02	2.28	20.34	0.93	78.49	3.64	86.84	5.39
60	59.25	2.75	98.37	0.15	40.80	1.54	42.07	2.12	20.30	0.93	78.76	3.23	87.89	5.07
70	59.96	2.45	98.46	0.16	43.15	1.24	41.97	2.02	20.03	0.93	79.19	2.98	87.81	4.63
80	60.73	2.37	98.50	0.14	44.78	1.46	41.77	1.77	20.19	0.86	79.30	2.70	88.60	4.57
90	61.27	2.06	98.56	0.10	46.27	1.60	41.75	1.72	20.17	0.85	79.68	2.93	89.74	3.42
100	61.98	2.08	98.59	0.11	47.48	1.63	41.67	1.87	20.12	0.76	79.89	3.22	90.09	3.94
150	64.35	1.68	98.69	0.10	50.86	1.44	41.50	1.92	20.26	0.68	80.22	2.63	91.32	3.79
200	66.00	1.09	98.75	0.10	52.80	1.25	41.41	1.36	20.04	0.67	80.22	2.70	90.70	4.12
250	67.01	0.90	98.79	0.08	54.01	1.01	41.38	1.42	20.09	0.62	80.16	2.43	90.53	3.37

Table 5: Average testing time (in seconds)

Dim. (k)	IRMA	COIL-100	ZuBuD	MIAS	Outex	Colon	Leukemia
5	71	46	0.93	0.11	0.44	0.01	0.01
10	137	87	1.76	0.20	0.81	0.02	0.01
15	207	129	2.57	0.27	1.17	0.03	0.01
20	278	172	3.39	0.37	1.56	0.02	0.02
25	344	216	4.20	0.45	1.94	0.02	0.01
30	404	258	5.01	0.53	2.31	0.03	0.01
35	478	339	5.84	0.61	2.69	0.03	0.01
40	541	344	6.65	0.70	3.08	0.03	0.01
45	609	388	7.44	0.78	3.57	0.04	0.01
50	676	433	8.25	0.87	3.86	0.04	0.01
60	809	517	9.91	1.05	4.55	0.05	0.02
70	941	617	11.57	1.23	5.44	0.05	0.02
80	1073	698	13.20	1.38	6.48	0.06	0.03
90	1206	770	14.82	1.56	6.79	0.06	0.03
100	1429	855	16.63	1.76	7.59	0.07	0.03
150	1998	1275	24.60	2.55	11.36	0.10	0.04
200	3279	1698	32.87	3.45	15.02	0.14	0.05
250	3354	2175	41.47	4.33	18.75	0.17	0.07
All	13399	8618	168.23	15.50	72.70	1.29	1.77

Emergence of ontological relations from visual data with Self-Organizing Maps

Jorma Laaksonen and Ville Viitaniemi
*Helsinki University of Technology
Adaptive Informatics Research Centre
FI-02015 TKK, FINLAND
jorma.laaksonen@tkk.fi ville.viitaniemi@tkk.fi

Abstract

In this paper we examine how Self-Organizing Maps (SOMs) can be used in detecting and describing emergent ontological relations between semantic objects and object classes in a visual database. The ontological relations we have studied include *co-existence*, *taxonomies of visual and semantic similarity* and *spatial relationships*. The used database contains 2618 images, each of which belongs to one or more of ten predefined semantic classes.

1 Introduction

Data-driven learning of semantic properties and ontological relations has for long been a goal in AI research. If machine learning were practicable, rule-based generation and presentation of *a priori* top-down knowledge could be complemented by information obtainable from bottom-up data processing and abstraction.

Techniques of traditional statistical pattern recognition and soft-computing, including neural networks and fuzzy logic, have been used for data-driven learning, but they have two severe limitations. First, they need training data, and supervised classifiers additionally *a priori* knowledge of correct classifications of the data. Second, statistical classifiers and other non-structural recognizers mostly operate on very low abstraction level data and there will inevitably exist the *semantic gap* between their output and and the required input for syntactic and other structural processing stages.

In this paper, we will study a specific image collection of 2618 images. The images are accompanied with keyword-type annotations which specify a subset of ten available keywords for each image. In addition, there exists information on the location of the objects in the images in the form of bounding boxes. The image database has been prepared for evaluation of object recognition and detection techniques, but it lends itself also for the purpose of studying the structure and ontological properties of the contained images.

The experiments to be described have been performed by using the PicSOM image analysis, classification and content-based information retrieval (CBIR) framework (Laaksonen et al., 2001, 2002). In the PicSOM system, we have recently added functionality for processing and analyzing also images which have hierarchical segmentations (Viitaniemi and Laaksonen, 2006b). As its name suggests, PicSOM is based on using Self-Organizing Maps (SOMs) (Kohonen, 2001) in organizing and indexing objects of a database. We believe that SOMs – due to their un-supervised training – can be a beneficial tool for facilitating data-driven emergence of ontological relations.

The rest of this paper is organized as follows. First, in Section 2, a short literature review is presented. Then, we give overviews of the Self-Organizing Map, the PicSOM CBIR system and the low-level visual features used in it in Section 3. Next, in Section 4, we describe the image data set used in the study. Section 5 presents and demonstrates the techniques we have developed for using SOMs in discovering and analyzing ontological relations in an image database. Conclusions are drawn and future directions discussed in Section 6.

2 Related work

Ontologies can be used to boost the performance of information retrieval systems (e.g. Bodner and Song (1996)). This applies to both large external general purpose ontologies, such as the WordNet (Fellbaum, 1998) with over one hundred thousand words, as well as ontologies mined automatically from the data itself. In the visual domain, ontologies have been used e.g. in aiding automatic image annotation (Jin et al., 2005; Srikanth et al., 2005), keyword sense disambiguation (Benitez and Chang, 2002), video retrieval (Hoogs et al., 2003; Hollink and Worring, 2005), and video annotation (Bertini et al., 2005).

For the ontologies to be usable in connection with visual data, they need to include visual components. In the literature, this has been achieved mostly by two means. First is linking pre-existing ontologies, such as the WordNet (Hoogs et al., 2003; Hollink and Worring, 2005; Jaimes and Smith, 2003) or domain ontologies, with visual concepts. The visual concepts are chosen so that they can be detected automatically from image or video documents. "Sky", "water" and "man-made" are examples of such concepts. In (Bertini et al., 2005) the concepts obtained by clustering visual features of manually chosen representative video clips of soccer game highlights are linked directly to the soccer game domain ontology nodes as specializations. Also in other approaches the linking involves manual labor. An advantage of this method is the ability to use pre-existing ontologies that are possibly very large. However, if the ontology is large and there are only a few visual concepts, the coupling between specific ontology concepts and visual properties may remain weak.

Another approach for building visual ontologies is to hierarchically cluster a visual training data set (e.g. Khan and Wang (2002)). Such clustering methods (e.g. Frakes and Baeza-Yates (1992)) have been used earlier for ontology discovery in textual and numerical data (Bodner and Song, 1996; Clerkin et al., 2001). An alternative to clustering the training data itself is to calculate a similarity measure between the ontology concepts on basis of the training data, and cluster directly the concepts based on the measure. In the clustering approach, labels of the ontology nodes are pre-specified by labels of the training data, and the relations between the nodes are learned. The number of nodes varies from a few to few dozens. (Khan and Wang, 2002) performs the clustering on basis of weighted cosine distance measure between the region composition vectors of images. In (Koskela and Smeaton, 2006) the distances between concepts describing videos are measured by the Jeffrey's divergence between their distributions in quantized feature spaces. In this work, we will consider measuring the distance between concepts by the performance of a visual classifier attempting to separate the concepts.

Describing spatial relationships in images has attracted research attention for decades (e.g. Winston (1975)). Qualitative descriptions based on logical formalisms have often been used. More recently, fuzzy descriptions of spatial relationships (Miyajima and Ralescu, 1994; Bloch, 2005) have become popular in the field. Spatial data mining from large databases (e.g. Koperski et al. (1996)) is a relatively new development. (Smith and Bridges, 2002; Lan et al., 2005) exemplify data mining based on fuzzy spatial relationships.

In this paper, we will use Self-Organizing Maps for the discovery and visualization of ontological relations—including spatial relationships—using visual data. The SOM has been used for ontology discovery and visualization earlier by Elliman and Pulido (2001, 2002) with textual and categorical data.

3 SOMs and PicSOM

3.1 Self-Organizing Map

The Self-Organizing Map (SOM) is an unsupervised, self-organizing neural algorithm widely used to visualize and interpret large high-dimensional data sets. The SOM defines an elastic net of points that are fitted to the distribution of the data in the input space. It can thus be used to visualize multi-dimensional data, usually on a two-dimensional grid.

The SOM consists of a two-dimensional lattice of neurons or map units. A model vector $\mathbf{m}_i \in \mathbb{R}^d$ is associated with each map unit i. The map attempts to represent all the available observations $\mathbf{x} \in \mathbb{R}^d$ with optimal accuracy by using the map units as a restricted set of models. During the training phase, the models become ordered on the grid so that similar models are close to and dissimilar models far from each other.

When training a SOM, the fitting of the model vectors is carried out by a sequential regression process, where $t = 0, 1, 2, \ldots, t_{max} - 1$ is the step index: For each input sample $\mathbf{x}(t)$, first the index $c(\mathbf{x})$ of the best-matching unit (BMU) or the "winner" model $\mathbf{m}_{c(\mathbf{x})}(t)$ is identified by the condition

$$\forall i: \quad \|\mathbf{x}(t) - \mathbf{m}_{c(\mathbf{x})}(t)\| \leq \|\mathbf{x}(t) - \mathbf{m}_i(t)\| \ . \quad (1)$$

The distance metric used here is usually the Euclidean one. After finding the BMU, the vectors of map units constituting a *neighborhood* centered around the node $c(\mathbf{x})$ are updated as

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + h(t; c(\mathbf{x}), i)(\mathbf{x}(t) - \mathbf{m}_i(t)) \ . \quad (2)$$

Here $h(t; c(\mathbf{x}), i)$ is the neighborhood function, a decreasing function of the distance between the ith and

$c(\mathbf{x})$th nodes on the map grid. This regression is reiterated over the available samples and the value of $h(t; c(\mathbf{x}), i)$ is let decrease in time to guarantee convergence of the model vectors $\mathbf{m}_i$. Large values of the neighborhood function $h(t; c(\mathbf{x}), i)$ are used in the beginning of the training for initializing the network, and small values on later iterations are needed for fine-tuning. After the training, any vector in the feature space can be quantized to a two-dimensional index by its BMU on the SOM.

3.2 PicSOM

In the PicSOM[1] image analysis framework, target objects are ranked according to their similarity with a given set of positive example objects, simultaneously combined with the dissimilarity with a set of negative example objects. The objects are correlated in terms of a large set of visual features of statistical nature. For this purpose training and test set images are preprocessed in the same manner: the images are first automatically segmented and a large body of statistical features is extracted from both the segments and whole images. Several different segmentations of the same images can be used in parallel. In the current experiments we consider only visual features, but the framework has been used to index also e.g. videos and multimedia messages (Koskela et al., 2005).

After feature extraction, a SOM is trained in an unsupervised manner to quantize each of the formed feature spaces. The quantization forms representations of the feature spaces, where points on the SOM surfaces correspond to images and image segments. Due to the topology preserving property of SOM mapping, the classification in each of the individual feature spaces can be performed by evaluating the distance of representation of an object on the SOM grid to the representations of positive and negative example objects.

The positive and negative examples are marked on the SOM surfaces with small impulse values of corresponding sign. As the SOM maps similar objects in nearby map units, we are motivated to spatially spread these sparse values fields by a low-pass filter, i.e. to convolve them with a smoothing kernel. The size and shape of the convolution kernel is selected in a suitable way in order to produce a smooth value map. In the resulting map, each location is effectively assigned a relevance value according to the number of positive objects mapped to the nearby units.

A single combined similarity measure is formed by summing the contributions of the individual feature spaces. In the classification task where the target objects are images, the segment-wise similarities are finally combined within an image by summing the contributions of all the segments in the image.

3.3 Low-level visual features

The PicSOM system implements a number of methods for extracting different statistical visual features from images and image segments. These features include a set of MPEG-7 content descriptors (ISO/IEC, 2002; Laaksonen et al., 2002) and additionally some non-standard descriptors for color, shape and texture.

3.3.1 Color

Of the used MPEG-7 descriptors Color Layout, Dominant Color and Scalable Color describe the color content in image segments. In addition to the MPEG-7 color descriptors, both the average color in the CIE L*a*b* color space (CIE, 1976) and three first central moments of the color distribution are used as color features.

3.3.2 Shape

Besides the MPEG-7 Region Shape, the shape features include two non-standard descriptors. The first consists of the set of the Fourier descriptors for the region contour. Fourier descriptors are derived from the following expansion of the region contour:

$$z(s) = \sum_{n=-\infty}^{\infty} z_n e^{\frac{2\pi i n s}{L}}. \qquad (3)$$

Here the Cartesian coordinates of the contour are represented by the real and the imaginary parts of the complex function $z(s)$, parameterized by the arc length s. The resulting feature vector includes a fixed number of low-order expansion coefficients z_n. The coefficients are then normalized against affine image transformations. In addition, the high-order coefficients are quadratically emphasized.

The second non-standard shape descriptor is formed from the Zernike moments (Khotanzad and Hong, 1990) of the region shape. The Zernike polynomials are a set of polar polynomials that are orthogonal in the unit disk. The Zernike moments A_{nm} are given by the expansion coefficients when the polar presentation of the region shape is represented in the basis of Zernike polynomials:

$$A_{nm} = \frac{n+1}{\pi} \sum_x \sum_y I_{xy} V_{nm}\left(\rho_{xy}, \theta_{xy}\right), \qquad (4)$$

[1] http://www.cis.hut.fi/picsom

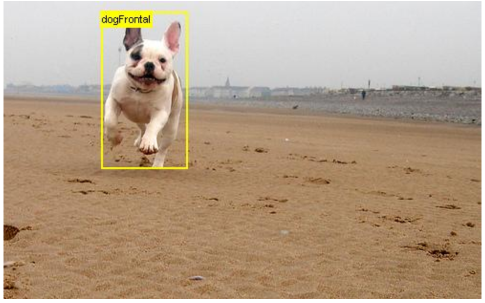


Figure 1: A sample image with keyword *dog* and its bounding box

where $n - |m|$ is even. Here n is the order of the moment, m the index of repetition, x, y are the rectangular image coordinates, and ρ_{xy}, θ_{xy} the corresponding polar coordinates. I_{xy} is the binary representation of the region shape and V_{nm} is the Zernike polynomial:

$$V_{nm}(\rho, \theta) = R_{nm}(\rho) e^{im\theta} \tag{5}$$

$$R_{nm}(\rho) = \sum_{s=0}^{\frac{n-|m|}{2}} \frac{(-1)^s (n-s)! \rho^{n-2s}}{s!(\frac{n+|m|}{2} - s)!(\frac{n-|m|}{2} - s)!}. \tag{6}$$

The feature vector includes coefficients A_{nm} up to a selected order. The feature is normalized against translation and scaling by fitting the region inside the unit disk. Rotation invariance is achieved by taking the absolute values of the coefficients.

3.3.3 Texture

We have used MPEG-7's Edge Histogram descriptor to describe the statistical texture in image segments. For non-standard description of a region's texture the YIQ color space Y-values of the region pixels are compared with the values of their 8-neighbors. The feature vector describes the statistics of the resulting distribution.

4 VOC image database

The Visual Object Classes (VOC) 2006 image database was created and annotated with keywords and bounding box information for the purpose of a challenge evaluation of object recognition and detection techniques[2]. It contains 2618 PNG images whose sizes are typically 500×375 pixels in either portrait or landscape orientation.

Figure 1 displays one example from the collection. It has been given the annotation *dogFrontal* and the associated bounding box. In our experiments, however, we have removed all pose information from the annotations and therefore used only the keyword *dog* for that image.

After discarding the pose information, there were ten keyword classes left in the database. The classes and the numbers of images in each of them are listed in Table 1. One may note that the total number of keywords is well above the number of images, which is an indication of the fact that many of the images have more than one keyword associated.

Table 1: The keyword classes in the VOC2006 database and their counts

bicycle	270
bus	174
car	553
cat	386
cow	206
dog	365
horse	247
motorbike	235
person	666
sheep	251

5 Ontological relations on SOMs

In this section we study three different types of information present in the VOC image data that can successfully be visualized and analyzed on a SOM.

5.1 Co-existence of classes

First we note that the ten keywords defined for the images in the collection form a ten-dimensional binary space and that each image is mapped to one point of that space. The ordering of the classes in such a vectorial presentation is meaningless, but the simultaneous existence of more than one keyword in one image is of special interest here.

For ten keywords, the *binary keyword vector* can have $2^{10} = 1024$ different combinations or values. The distribution of vectors in this ten-dimensional space will thus be difficult to visualize without means for low-dimensional projections. We have used SOM for this purpose as follows.

First, we have added small amount of white Gaussian noise in the binary vectors. This ensures that

[2] `http://www.pascal-network.org/challenges/VOC/`

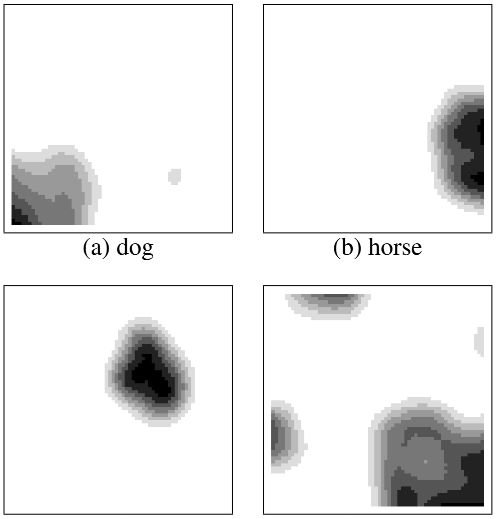
there will not exist two exactly equal data vectors even though two images may have the same associated keywords. As a further consequence, those keyword combinations which are most common in the image collection will occupy largest areas on the SOM to be created.

Second, we trained a SOM and studied for each of the ten keyword classes how images that had been given that keyword are mapped on the keyword SOM. Figure 2 shows the SOM areas where four image classes are mapped. It can be seen that the class *person* intersects notably with both *dog* and *horse*, but not at all with *sheep* in this database.

After inspecting the distributions of all the ten keyword classes, one can depict a *co-existence graph* for describing the ontological relations of the classes with respect to the likeliness of simultaneous existence in a photograph. The areas on the SOM surface can be used to guide the placement of the nodes in the graph, as can be seen done in Figure 3. It is obvious that in this collection the *person* class is very central as almost all the other classes exhibit co-existence with it. Furthermore, all the different vehicles form a tightly interconnected subgraph.

5.2 Visual and semantic taxonomy

With our second experiment we show that the visual similarities between the classes can be used in data-driven creation of a taxonomy of object classes. For that purpose, we first calculated pairwise misclassification rates between all class pairs. That is, we assessed the similarity of two classes by studying the difficulty of separating them. For achieving optimal classification we used a k-nearest neighbor classifier with optimization of the k value and sequential forward selection (SFS) of the used visual features.


(a) dog

(b) horse

(c) sheep

(d) person

Figure 2: Areas occupied by four different image classes on keyword SOM

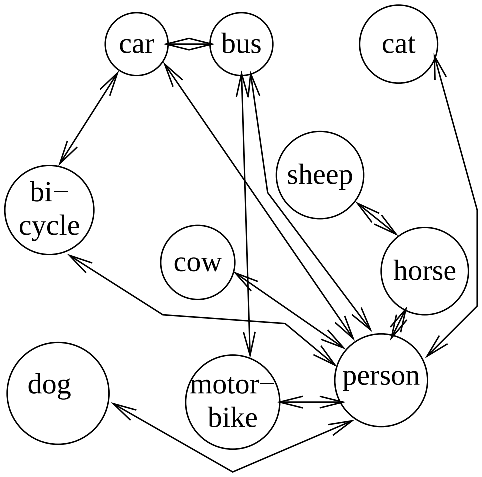


Figure 3: Co-existence graph of the VOC image collection, created to match corresponding class areas on the keyword SOM

When the two classes which were the most difficult to separate were found, we created a virtual joint class of those two and re-calculated all pairwise misclassification rates. This process produced the unbalanced binary tree of the ten classes shown in Figure 4. We argue that even though this *visual class taxonomy* has been created purely from low-level visual feature data, it to some extent coincides with the corresponding *semantic class taxonomy* one might come to by mental contemplation.

In the VOC data, there exist class pairs (*motorbike, bicycle*), (*cat,dog*) and (*cow,sheep*) which tend to be misclassified and are therefore combined in the early stages of the taxonomy tree formation process of Figure 4. Some of the later merges are, however, not as straightforward to interpret. (One may actually state that such an iterative merging process which only relies on local information is bound to be chaotic.)

The same principle can be used also to create a one-dimensional, non-hierarchical ordering of the classes where mutually similar ones reside near each other. In order to achieve this, we first changed for each class the cardinal numbers that showed the misclassification rate between that class and each other class to ordinal numbers which indicated the order of the other classes in decreasing misclassification rate.

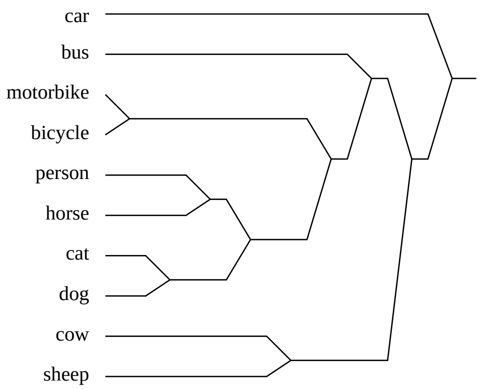


Figure 4: Taxonomy tree of visual similarity of the classes

Then we re-ordered the rows and columns of the resulting matrix so that the smallest numbers we forced to reside along the diagonal and the first subdiagonals whereas the largest numbers were moved to furthest away from the diagonal. The resulting ordering and *ordinal similarity matrix* is shown in Figure 5. It can be seen that there again exist obvious pairs (*motorbike, bicycle*), (*cat,dog*) and (*cow,sheep*), but they are now aligned on a linear continuum between the two opposites, *bus* and *sheep*.

The creation of the two similarity representations for the ten object class were based on classifications which used potentially all the available low-level features. Therefore, there does not exist a single SOM map, where the ordering of the object classes would match that of the class taxonomy tree or the ordered similarity matrix. Some features, however, may come quite close to that ordering. For an illustration, see Figure 6 where the VOC image dataset has been ordered on a 16×16-sized SOM created from the MPEG-7 EdgeHistogram feature. One can see how the antidiagonal ordering of the classes *car–bus–motorbike–bicycle–horse–dog/cat–sheep–cow* quite well matches those of Figures 4 and 5.

bus	0	3	2	4	1	5	6	7	9	8
car	3	0	2	4	1	5	6	7	9	8
motorbike	4	5	0	1	2	3	6	7	9	8
bicycle	7	6	1	0	2	4	5	3	8	9
person	7	6	1	3	0	2	5	4	9	8
horse	9	8	3	6	1	0	3	2	5	7
cat	8	9	5	4	3	2	0	1	5	7
dog	8	9	7	6	3	2	1	0	5	4
cow	8	8	7	6	5	2	4	3	0	1
sheep	9	8	5	7	4	3	6	2	1	0

Figure 5: Ordered similarity matrix of the object classes

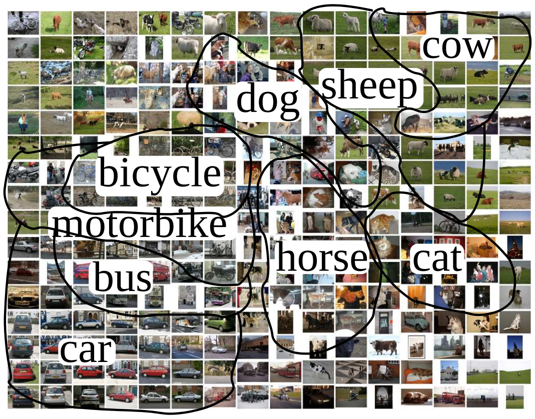


Figure 6: Ordering of object classes on MPEG-7 EdgeHistogram SOM

5.3 Spatial relationships

With our final experiment we want to show how the SOM can be used in presenting spatial relationships between objects in an image. We will here restrict our attention to images where there exists one *horse* and one *person*. The database contains a total of 84 such images. From the associated annotations we can extract the bounding boxes of those two objects and combine them in eight-dimensional feature vectors.

Figure 7 displays a 16×16-sized SOM created from the eight-dimensional spatial location data. One may coarsely identify four regions on the map. First, on the left of the map there are images where the person is standing to the right of the horse. On the bottom left the situation is otherwise similar, but the spatial ordering of the objects is reversed. On the right of the map there are portrait images of riding and in the middle parts similar landscape images.

We argue that the *spatial relationship SOM* has been able to discover relations between the horses and persons which we would associate to “riding” and “standing”. This is an example of data-driven emergence by a SOM which orders objects topographically on the basis of their low-level statistical features. Only the linguistic concepts “riding” and “standing” have in this case been added by human inspection of the outcome of the self-organization.

In this example we have used only data from images where *horse* and *person* have been the two objects whose spatial relationships have been of interest.

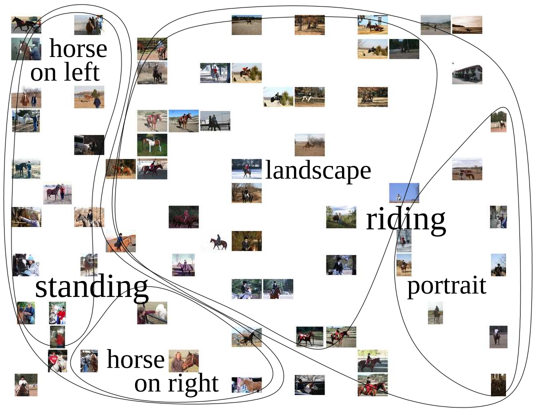


Figure 7: Spatial relationship SOM created from images of a horse and a person

The approach is, however, independent of the classes, so a single SOM could be created from data of all object pairs. Such a map will inevitably describe the relationships on a more general level, for example, the relationship of *person* "riding" *horse* would be replaced by a less context-specific expression *person* "above" *horse*.

6 Conclusions and discussion

In this paper we have shown that the Self-Organizing Map can successfully be applied in diverse data analysis tasks in which it is able to extract and display emergent ontological relations between semantic objects and object classes. The studied ontological relations included (i) simultaneous existence of objects from two or more object classes in one image, (ii) taxonomy of visual (and arguably to some extent also semantic) similarity, and (iii) spatial relationships between different object types in one image.

The experiments were performed within the PicSOM framework for image analysis and content-based information retrieval. The PicSOM system inherently uses SOMs in indexing and scoring images by their similarity to other images known to be relevant in a particular query instance. Consequently, the co-existence and spatial relationship SOMs, which were for the first time introduced in this paper, can easily be integrated in that framework.

In another related study (Viitaniemi and Laaksonen, 2006a) we have already questioned whether knowledge of a semantic taxonomy of object classes would be beneficial for detection of objects. This is an important issue because semantically related object types often appear in visually similar *contexts*. Therefore hierarchical detection of such objects may to some level rely also on the context information, but on the final stages the visual appearance of the sole objects itself must be the only discriminating factor as the context will be equally probable for all the candidate object classes.

In the future we will turn our attention to more quantitative analysis and assessment of the benefits obtainable from the techniques presented in this paper. In the forthcoming experiments we will study how the emergent ontological relations can be first extracted from available annotated training data and then be used to facilitate efficient retrieval from unannotated test data.

Acknowledgments

This work has been supported by the Academy of Finland in the projects *Neural methods in information retrieval based on automatic content analysis and relevance feedback* and *Finnish Centre of Excellence in Adaptive Informatics Research*.

References

Ana B. Benitez and Shih-Fu Chang. Semantic knowledge construction from annotated image collections. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME 2002)*, volume 2, pages 205–208, Lausanne, Switzerland, August 2002.

Marco Bertini, Alberto Del Bimbo, and Carlo Torniai. Automatic video annotation using ontologies extended with visual information. In *ACM Multimedia*, pages 395–398, 2005.

Isabelle Bloch. Fuzzy spatial relationships for image processing and interpretation: a review. *Image and Vision Computing*, 23:89–110, February 2005.

R. Bodner and F. Song. Knowledge-based approaches to query expansion in information retrieval. In G. McGalla, editor, *Advances in Artificial Intelligence*, pages 146–158. Springer, New York, 1996.

CIE. Supplement No. 2 to CIE publication No. 15 Colorimetry (E-1.3.1) 1971: Official recommendations on uniform color spaces, color-difference equations, and metric color terms, 1976.

P. Clerkin, P. Cunningham, and C. Hayes. Ontology discovery for the semantic web using hierarchical clustering. In *Online proceedings of Semantic Web Mining Workshop at*

ECML/PKDD-2001, Freiburg, Germany, 2001. http://semwebmine2001.aifb.uni-karlsruhe.de/.

D. Elliman and J. R. G. Pulido. Visualizing ontology component through Self-Organizing Maps. In *Proceedings of the International Conference on Information Visualisation (IV2002)*, pages 434–438, London, July 2002.

Dave Elliman and J. Rafael Pulido. Automatic derivation of on-line document ontologies, 2001.

Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. The MIT Press, 1998.

William B. Frakes and Ricardo Baeza-Yates, editors. *Information Retrieval, Data Structures and Algorithms*. Prentice Hall, New Jersey, USA, 1992.

Laura Hollink and Marcel Worring. Building a visual ontology for video retrieval. In *ACM Multimedia*, pages 479–482, 2005.

Anthony Hoogs, Jens Rittscher, Gees Stein, and John Schmiederer. Video content annotation using visual analysis and a large semantic knowledgebase. In *CVPR (2)*, pages 327–334, 2003.

ISO/IEC. Information technology - Multimedia content description interface - Part 3: Visual, 2002. 15938-3:2002(E).

Alejandro Jaimes and John R. Smith. Semi-automatic, data-driven construction of multimedia ontologies. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, Baltimore, July 2003.

Yohan Jin, Latifur Khan, Lei Wang, and Mamoun Awad. Image annotations by combining multiple evidence & WordNet. In *ACM Multimedia*, pages 706–715, 2005.

Latifur Khan and Lei Wang. Automatic ontology derivation using clustering for image classification. In *Multimedia Information Systems*, pages 56–65, Tempe, AZ, USA, October 2002.

A. Khotanzad and Y. H. Hong. Invariant image recognition by Zernike moments. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 12(5): 489–497, 1990.

Teuvo Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer-Verlag, Berlin, third edition, 2001.

Krzysztof Koperski, Junas Adhikary, and Jiawei Han. Spatial data mining: progress and challenges. In *Proceedings of the Workshop on Research Issues on Data Mining and Knowledge Discovery*, Montreal, Canada, 1996.

Markus Koskela and Alan F. Smeaton. Clustering-based analysis of semantic concept models for video shots. In *International Conference on Multimedia & Expo (ICME 2006)*, Toronto, Canada, July 2006. To appear.

Markus Koskela, Jorma Laaksonen, Mats Sjöberg, and Hannes Muurinen. PicSOM experiments in TRECVID 2005. In *Proceedings of the TRECVID 2005 Workshop*, pages 262–270, Gaithersburg, MD, USA, November 2005.

Jorma Laaksonen, Markus Koskela, Sami Laakso, and Erkki Oja. Self-organizing maps as a relevance feedback technique in content-based image retrieval. *Pattern Analysis & Applications*, 4(2+3): 140–152, June 2001.

Jorma Laaksonen, Markus Koskela, and Erkki Oja. PicSOM—Self-organizing image retrieval with MPEG-7 content descriptions. *IEEE Transactions on Neural Networks, Special Issue on Intelligent Multimedia Processing*, 13(4):841–853, July 2002.

Rongqin Lan, Wenzhong Shi, Xiaomei Yang, and Guangyuan Lin. Mining fuzzy spatial configuration rules: methods and applications. In *IPRS Workshop on Service and Application of Spatial Data Infrastructure*, pages 319–324, 2005.

Koji Miyajima and Anca Ralescu. Spatial organization in 2D images. In *Proceedings of the Third IEEE Confence on Fuzzy Systems*, volume 1, pages 100–105, 1994.

George Brannon Smith and Susan M. Bridges. Fuzzy spatial data mining. In *Proceedings of the 2002 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS2002)*, pages 184–189, 2002.

M. Srikanth, J. Varner, M. Bowden, and D. Moldovan. Exploiting ontologies for automatic image annotation. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 552–558, 2005.

Ville Viitaniemi and Jorma Laaksonen. Techniques for still image scene classification and object detection. In *Proceedings of 16th International Conference on Artificial Neural Networks (ICANN 2006)*, September 2006a. To appear.

Ville Viitaniemi and Jorma Laaksonen. *Focusing Keywords to Automatically Extracted Image Segments Using Self-Organising Maps*, volume 210 of *Studies in Fuzziness and Soft Computing*. Springer Verlag, 2006b. ISBN 3-540-38232-1. To appear.

P. H. Winston. Learning structural descriptions from examples. In P. H. Winston, editor, *The Psychology of Computer Vision*. McGraw-Hill, 1975.

Mutated Kd-tree Importance Sampling

Perttu Hämäläinen*, Timo Aila*, Tapio Takala*
*Helsinki University of Technology
Telecomm. Software and Multimedia Laboratory
{pjhamala,timo,tta}@tml.tkk.fi

Jarmo Alander†
†University of Vaasa
Dept. of Electrical Engineering and Automation
Jarmo.Alander@uwasa.fi

Abstract

This paper describes a novel importance sampling method with applications in multimodal optimization. Based on initial results, the method seems suitable for real-time computer vision, and enables an efficient frame-by-frame global search with no initialization step. The method is based on importance sampling with adaptive subdivision, developed by Kajiya, Painter, and Sloan in the context of stochastic image rendering. The novelty of our method is that the importance of each kd-tree node is computed without the knowledge of its neighbours, which saves computing time when there's a large number of samples and optimized variables. Our method can be considered a hybrid of importance sampling, genetic algorithms and evolution strategies.

1 Introduction

Model-fitting in real-time computer vision is often a difficult multimodal, multivariable optimization problem. It is common to only search a portion of the parameter space based on prior knowledge, such as a Kalman filter that predicts the next solution based on the previous solutions (Sonka *et al.*, 1999). In cases where the tracked shape moves in an unpredictable and rapid manner, e.g., when tracking the user in gesture-controlled software, the prediction may not be accurate. In this case, the tracker may need to be re-initialized.

From the point of view of usability, initialization and too strong priors should be avoided so that the user controls the software instead having to adapt to the limits of technology. For example, Pfinder initializes the tracking only if the user assumes a pose in which the body parts can be reliably labelled based on the user's silhouette (Wren *et al.*, 1997).

To avoid re-initialization, more efficient global optimization strategies need to be developed. This paper describes a novel optimization method that we call *mutated kd-tree importance sampling*. The method is a hybrid of importance sampling, genetic algorithms (GA) and evolution strategies (ES). It is based on the following design principles:

Principle 1: Optimization as sampling
To inspect all peaks of the objective function (fitness function) carefully without neglecting any part of the parameter space, the density of samples at a given location should be proportional to the fitness of the location. In other words, we inspect optimization as importance sampling, treating the fitness function as a probability distribution function.

Principle 2: Make every sample count
Evaluating the fitness of generated samples is often time-consuming. For maximum efficiency, the decision of where to place the next sample should be supported by all past samples. Each new sample increases the resolution of the perceived fitness function shape.

Principle 3: Simple user interface
An optimization method should have as few parameters as possible for it to be easy to use. However, parameters are necessary to incorporate problem-specific prior knowledge to the optimization. The No Free Lunch (NFL) theorems for optimization state that on average, all optimization methods are equal when applied to all possible cost functions without problem-specific adjustments (Wolpert and MacReady, 1997).

Based on the design principles, we have developed a novel variant of kd-tree based importance sampling introduced by Kajiya (1986) and developed further by Painter and Sloan (1989).

2 Description of the method

Our method uses a kd-tree to adaptively subdivide the search space. With *N* optimized variables, each node of the tree represents an *N*-dimensional hypercube in the search space. Each node has two children. The children are created by bisecting the parent along a coordinate axis. There's one sample inside each leaf node.

Samples are generated so that a leaf node is sampled from a discrete probability distribution where the relative probability of leaf node *k* equals a single point estimate of the integral of the fitness inside the node, $p_k=f_k v_k$, where f_k is the fitness of the sample inside the node and v_k is the volume of the node.

Contrary to previous kd-tree importance sampling methods, we do not generate a sample uniformly inside the selected hypercube, which would correspond to using the kd-tree as a piecewise constant approximation of the fitness function. Instead, we mutate the existing sample of the selected hypercube using a multivariate normal distribution centered at the sample or the hypercube center. The covariance matrix is diagonal and the standard deviations are proportional to the dimensions of the hypercube. This way, the kd-tree approximates the fitness function as an additive mixture of Gaussians.

A new Gaussian is created for each mutated sample by finding the leaf node (hypercube) inside which the sample is located, and splitting the node into two new hypercubes, one containing the old sample and the other containing the mutated one. In optimized kd-trees for databases, the splitting dimension is often chosen to be that whose distribution exhibits the most spread (Yianilos, 1993). In our case this corresponds to the dimension of maximum distance between the old and new sample.

The Gaussians could also be stored in some other data structure than a kd-tree. The benefit of using a kd-tree is that the node selection time grows logarithmically as a function of the number of samples. After a new sample is stored in a leaf node, the probabilities are updated recursively by traversing from the leaf to the root and setting $p_{parent}= p_{child1} + p_{child2}$. When selecting a leaf, the tree is traversed from the root to a leaf so that at each node, a uniformly distributed pseudorandom number *r* is generated in the range $0 \ldots p_{child1}+p_{child2.}$ Child 1 is visited if $r< p_{child1}$. If $r= p_{child1}$, the child is selected randomly.

The sampling is not prone to getting stuck inside a local optimum. A sample with high fitness attracts more samples, and if a sample belongs to a peak much smaller than the cube containing the sample, the fitness function approximation is inaccurate and the peak attracts disproportionately many samples. However, as the cubes get split to smaller and smaller ones, the approximation gets more accurate.

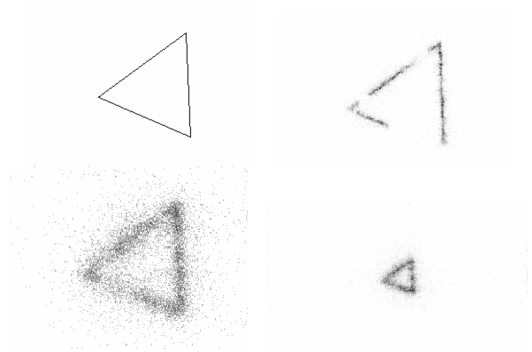


Figure 1. Top-left: 2d fitness function. Top-right: the distribution of samples with low mutation variance. Bottom-left: the distribution of samples with high mutation variance. Bottom-right: decreasing the high-fitness area reduces blurring.

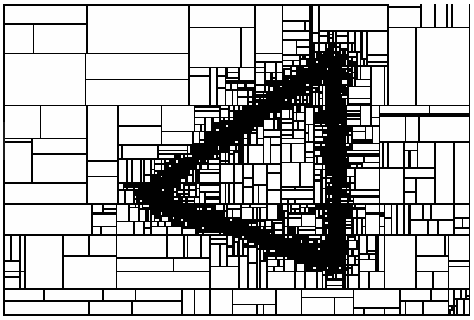


Figure 2. An example of the adaptive subdivision of space.

The normally distributed mutations provide a crucial improvement compared to a piecewise constant approximation of the fitness function: The spreading of samples beyond node boundaries ensures that a node can get split even if the fitness of its sample is initially zero ($p_k=f_k v_k=0$). The normally distributed sampling propagates the selection probabilities so that if the fitness of a cube is high, it will attract more samples both inside it and its neighbours.

Figure 1 illustrates the effect of the mutation variance. The 2d fitness function is a 256x256 pixel bitmap, sampled using 10000 samples (0.16 samples per pixel). The fitness function is zero except along the edges of the triangle, so that there's no gradient information to guide the search. With low variance, samples are highly concentrated on the triangle edges, but regions of the space remain unexplored due to zero samples. Increasing the mutation variance adjusts the compromise between greediness and thoroughness of the search. With high variance, the whole space gets searched, but the samples also fall on the zero areas due to a Gaussian blurring effect.

Thanks to the mutation variance proportional to the dimensions of the selected hypercube, the blurring is adaptive considering the number of samples and the size of the high-fitness area. The bottom-right image in Figure 1 shows how there's less blurring when the high-fitness area is smaller. The mutation variances at the bottom-left and bottom-right images are equal.

Figure 2 illustrates the adaptive subdivision resulting from the sampling.

3 Related work

3.1 Genetic algorithms

In multimodal, multivariable optimization, stochastic methods are generally a safer choice than hill-climbing methods that use the gradient of the objective function to direct the search. The hill-climbing (or descent) methods are prone to ending up in a local optimum instead of the global one.

Genetic algorithms (GA) are a vast family of stochastic optimization methods. GA methods generate samples (individuals) and evaluate their fitness. After an initial population has been generated by random sampling or based on prior knowledge, the population is evolved, which consists of selection, cross-over and mutation operations (e.g., Goldberg, 1989). In model-fitting, fitness can be formulated as a function of fitting error so that smaller error yields greater fitness. The optimized variables are formulated into a genome, e.g., a binary string or a vector of real values.

Our sampling method is related to GA in that, the corners of a kd-tree node can be thought as parents that produce offspring samples by cross-over. The structured subdivision provides the benefit of knowing that there's only one previous sample in the same space as the offspring. Even if the parents are fit, they should not be selected for cross-over if there already are several samples in the same space as the offspring. In this case, the offspring does not provide much additional information about the fitness function shape.

3.2 Evolution strategies

Evolution strategies (ES) are another big family of stochastic optimization methods. In contrast to GA, ES view the optimized variables not as single genes but as the features that are function of several genes, e.g., musical or mathematical talent (e.g., Beyer, 2001). Such features are often normally distributed in real populations. ES exploit this, e.g., by generating samples by sampling from a normal distribution centered at a selected individual.

Our method is related to ES in that we also select a sample and mutate it using a normal distribution. Our main difference to ES is the structured subdivision of space. The variance of the normal distribution is relative to the size of the selected hypercube, which focuses the search based on how much resolution there already is in fitness function approximation.

3.3 Importance sampling

Population based optimization can be sometimes replaced by importance sampling, and vice versa, although importance sampling is perhaps more typically used in Monte Carlo integration, i.e., estimating integrals of difficult integrands based on random samples. There are various importance sampling methods, of which overviews can be found in textbooks (e.g., Dutré, 2003). Borrowing ideas from computational chemistry, Sminchisescu (2002) has investigated hyperdynamic importance sampling for computer vision model-fitting. In computer graphics, both genetic algorithms and importance sampling have been used to minimize the number of light paths that need to be evaluated in image rendering (Szirmay-Kalos, 1999).

A simple way to estimate a definite integral is to compute the mean of random samples of the integrand within the integrating bounds. However, the estimate has a high variance, and much of the literature is devoted to variance reduction techniques.

Variance can be reduced by importance sampling, which means that more samples are produced in areas that contribute more to the integral. The relation to optimization is clear when considering a fitness function as the integrand: Areas of high fitness contribute more to the integral, and concentrating samples there makes it more probable to generate a sample close to the optimum.

Importance sampling is also related to stochastic optimization in that simulated annealing is an adaptation of Metropolis Monte Carlo (MMC) importance sampling. In MMC, a new sample is generated by displacing the previous one randomly. The change of the energy of the system, ΔE, is then computed and the new location is accepted if $\Delta E<0$ or if $\xi<\exp(-\Delta E/kT)$, where ξ is a random number between 0 and 1, and T is the temperature of the system. In the original Metropolis case, the sample is formed as a vector of the locations of the particles of a substance (Metropolis *et al.*, 1953).

In simulated annealing, energy is replaced by an arbitrary cost function, and T is gradually decreased so that the system freezes in a (near) optimal configuration (Kirkpatrick *et al.*, 1983). Although simulated annealing has been found powerful in many problems, it violates our principle 2 in that new samples are not based on all previous samples. The method is prone to get trapped inside a local optimum, especially if the temperature is decreased too rapidly. Variants of simulated annealing have been developed to improve transitions between optima, e.g., using gradient and curvature information to find saddle points representing ‘mountain passes’ connecting adjacent cost basins (Sminchisescu and Triggs, 2005). To our knowledge, none of the variants have exploited the adaptive subdivision of space that is characteristic to our method.

3.4 Sampling and adaptive subdivision using kd-trees

In Monte Carlo integration, variance can be reduced through stratification. This means that the sampling space is divided into equal size strata, and samples are generated inside each stratum. This ensures that the samples don’t get cluttered up in some portion of the space. Unfortunately, this is impractical in high-dimensional problems, because the number of strata grows exponentially as a function of dimensionality.

Crucial to this paper is the unification of importance sampling and stratification via adaptive subdivision using a kd-tree. This was first proposed by Kajiya (1986) in context of 3d image rendering. He writes that *“So far, our experiments in finding adaptive criteria have not been terribly successful”*, adaptive criteria denoting the means to decide which node to split and generate a new sample in. Although he outlined the use of the kd-tree to divide the space, he didn’t use the adaptive subdivision in generating the images in the paper.

Painter and Sloan (1989) were successful in developing the approach further. They were able to considerably reduce the number of light paths that need to be evaluated to provide an anti-aliased image.

Comparing Painter and Sloan to Kajiya, a major improvement is that the selection priority of a node equals the product of *external variance* and the volume of the node. External variance is computed from the means of the samples of the node and its neighbour nodes. This ensures that even if the priority of a node is initially zero, the priority grows along with the priorities of its neighbours and no part of the space is left unexplored. The effect is similar to the mutation in our method, but it is impractical in high-dimensional problems. The number of neighbours of a node grows exponentially as the number of dimensions increase. In high-dimensional spaces, updating the priorities may require thousands of operations for each new sample. This may be why Painter and Sloan apply their method in only two dimensions.

Compared to the randomized selection in our method (sampling from the approximated fitness distribution), Painter and Sloan always select the node with the highest priority. We also tried this, but it seems to increase the exploring of local optima and the time needed to find the global optimum. Randomized selection enhances the parallel exploration of several optima.

In our method, the concept of external variance can be implemented so that instead of fitness times volume, the relative probability of node k is computed as $p_k=|f_k\text{-}f_s|v_k$, where f_s is the fitness of the sibling of the node. This improves sampling of, e.g., images with areas of constant colour, but so far we have not observed a clear difference in optimization.

There’s been quite some time since Painter and Sloan’s paper, but the ideas have recently had revived interest. SUAVE, a method similar to Painter and Sloan’s has been implemented as a part of the CUBA library for multidimensional numerical integration (Hahn, 2005 and 2006). SUAVE improves previous methods by global estimation of Monte Carlo integration error and stops sampling automatically when desired precision has been reached.

4 Test results

Figure 1 demonstrates the ability of our method to concentrate samples in areas of high fitness, even when the gradient of the fitness function is either zero or infinite. In the following, we present the results from optimization in the case of two multidimensional test functions and a real-world computer vision problem with three optimized variables.

4.1 Multidimensional test functions

Figures 3 and 4 illustrate the average convergence of 100 test runs of our method with 1 to 10 dimensional test functions. In the figures, the Euclidian distance between the true optimum and the best sample so far is plotted as a function of samples generated. The distance curves are normalized so that they start at 1, independent of the number of dimensions. In Figure 3, the fitness is defined as

$$f(\mathbf{x}) = \prod_i \left|\mathrm{sinc}(x_i)\right|, \qquad (1)$$

where x_i denotes optimized parameter i. The parameter range is $-4\pi \le x_i \le 4\pi$. In Figure 4, the fitness is an exponential peak,

$$f(\mathbf{x}) = e^{-10\sqrt{|\mathbf{x}|}}. \qquad (2)$$

Because the samples are distributed according to the fitness, convergence is rapid if high fitness is concentrated close to the optimum, as in the case of the exponential peak. In general, exponentially peaked fitness functions work better than, e.g., quadratic peaks, since the convergence slows down if the gradient of the fitness approaches zero near the optimum.

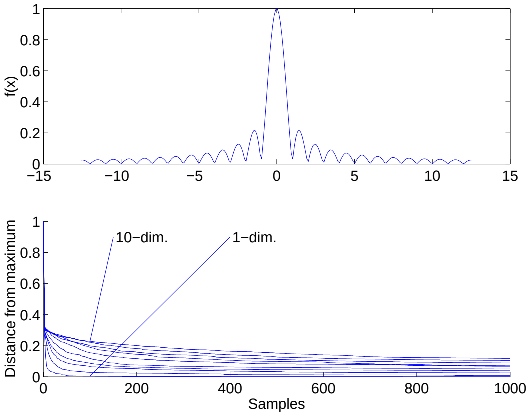


Figure 3. Convergence when finding the maximum of a multimodal test function in 1 to 10 dimensions.

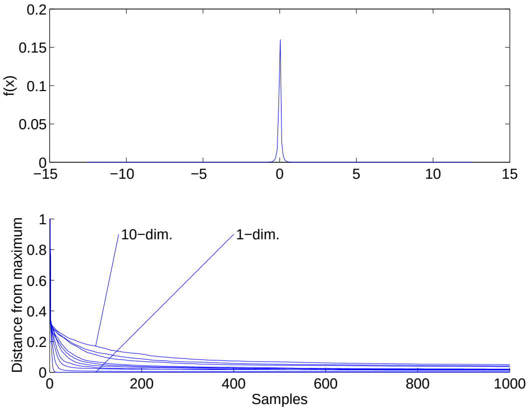


Figure 4. Convergence when finding the maximum of an exponential peak in 1 to 10 dimensions.

4.2 Exploiting spatiotemporal coherence in computer vision

Considering practical applications, we have tested our method in a recent version of a computer vision based user interface, originally presented by Mäki-Patola *et al.* (2006). Figure 5 shows the view of a camera placed inside a djembe drum. The shadows of the player's hands control musical synthesis. The hands don't need to be fully tracked, because moment-based image features provide enough information. However, the moments should only be computed at the drum membrane, for which a circular mask is needed. The mask is obtained by fitting a circle to the drum membrane, shown in Figure 5. This is an optimization problem with the circle coordinates and radius as the variables. The drum membrane stays mostly still in the camera view, but it may move when the drum is hit hard.

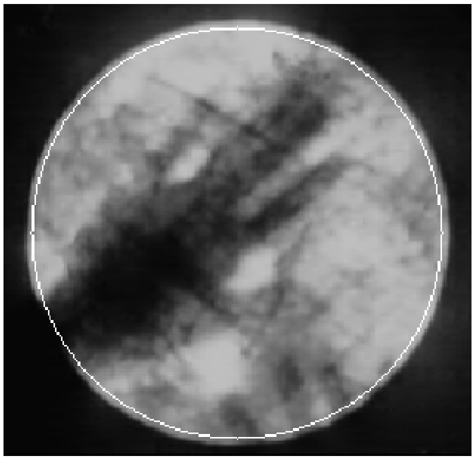


Figure 5. The view of a camera placed inside a djembe drum, showing the shadows of the player's hands on the drum membrane, and the circle fitted to the outline of the membrane.

We formulate the fitness function as

$$f = e^{-\sum_i I_{1i}} e^{-\sum_i (255 - I_{2i})} r , \qquad (3)$$

where I_{1i} denotes the intensity of pixel i of the circle, and I_{2i} denotes the intensity of pixel i of a circle with radius $1.05r$. The intensity values are in the range 0...255. The circle is sampled uniformly with 14 points. In a good solution, the circle is on the drum membrane, but a slightly larger circle falls outside the drum. Multiplying by r favours large circles.

According to our tests, 1000 samples are enough when optimizing the circle parameters at each new video frame with 30 frames per second. However, the performance can be boosted considerably by exploiting both spatial and temporal coherence.

The basic version of our sampling method exploits spatial coherence, that is, samples are likely to be generated in the vicinity of regions that are already known to yield high fitness. Temporal coherence can be utilized so that instead of initializing the optimization at each video frame, the kd-tree generated at the previous frame is pruned to only contain a number of best samples. The fitness of the samples is then re-evaluated in the context of the new video frame, after which optimization is continued normally. When preserving 10 best samples, we only need about 100 new samples for each frame. The computational and memory costs are considerably less than, e.g., those of Hough transform.

In effect, we initialize the optimization with guesses based on the previous optimization results, which is a common technique in computer vision. What makes our case special is that thanks to the adaptive subdivision and the sampling based on all previous samples, erroneous initial guesses do not prevent the optimization from converging. Sudden movements of the djembe only cause a few frames of confusion, after which the optimization converges again without any additional initialization.

5 Conclusion and future work

We have described a novel importance sampling method that improves previous methods by combining kd-tree adaptive subdivision with stochastic blurring of the sample distribution. The blurring is implemented via mutations that make it unnecessary to find and evaluate neighbours in the kd-tree. This can save computation time, especially in multi-parameter optimization.

The method satisfies all the three design criteria defined in the introduction: optimization as sampling from the fitness distribution, utilization of every past sample in the generation of a new sample, and a simple user interface. Besides the formulation of the fitness function, which typically requires skill and insight, the only user-adjustable parameters are the number of samples and the mutation variance that adjusts greediness. The sampling can also be initialized with a number of initial guesses, and it recovers from erroneous guesses.

We find the method promising especially in the following aspects that we are investigating for future publications:

- Light transport in image rendering. This is natural, as the roots of our method are in importance sampling of light paths.
- Improving the utilization of temporal coherence by incorporating ideas from particle filters, such as Condensation (Blake and Isard, 1998). It seems that using the kd-tree may ensure global sampling so that the particles don't get trapped inside a single mode of the estimated distribution.

Acknowledgements

We wish to thank all our colleagues for the fruitful discussions, especially Samuli Laine, Janne Kontkanen, Jaakko Lehtinen, and Teemu Mäki-Patola. This work has been partly funded by Academy of Finland.

References

Hans-Georg Beyer. *The Theory of Evolution Strategies*, Springer-Verlag, 2001

A. Blake and M. Isard. *Active Contours*. Springer-Verlag, 1998

P. Dutré, P. Bekaert, and K. Bala. *Advanced Global Illumination*. A K Peters Ltd., 2003

D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, 1989

T. Hahn. CUBA - a library for multidimensional numerical integration. *Computer Physics Communications*, 168:78–95, 2005

T. Hahn. The CUBA library. *Nuclear Instruments and Methods in Physics Research A*, 559:273–277, 2006

J.T. Kajiya. The rendering equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and interactive Techniques (SIGGRAPH '86)*, 143-150, 1986

S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by Simulated Annealing, *Science*, 220(4598): 671-680, 1983

N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equation of State Calculations by Fast Computing Machines, *The Journal of Chemical Physics*, 21(6): 1087-1092, 1953

T. Mäki-Patola, P. Hämäläinen, and A. Kanerva. The Augmented Djembe Drum – Sculpting Rhythms. To appear in *Proceedings of NIME'2006*, Ircam, France, June 5-8, 2006

J. Painter and K. Sloan. Antialiased ray tracing by adaptive progressive refinement. In *Proceedings of the 16th Annual Conference on Computer Graphics and interactive Techniques (SIGGRAPH '89)*, 281-288, 1989

C. Sminchisescu. *ESTIMATION ALGORITHMS FOR AMBIGUOUS VISUAL MODELS Three Dimensional Human Modeling and Motion Reconstruction in Monocular Video Sequences*, doctoral thesis, 2002, http://www.cs.toronto.edu/~crismin/thesis.html, link visited 29th April 2006

C. Sminchisescu and B. Triggs. Hyperdynamic Sampling. *Journal of Image & Vision Computing*, 2005, available online at http://lear.inrialpes.fr/pubs/2005/ST05a, link visited 29th April 2006

M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis and Machine Vision*, 2nd edition, Brooks/Cole Publishing Company, 1999

L. Szirmay-Kalos. *Monte-Carlo Methods in Global Illumination*. Institute of Computer Graphics, Vienna University of Technology, Vienna, 1999. http://citeseer.ist.psu.edu/szirmay-kalos00montecarlo.html, link visited 29th April 2006

D.H. Wolpert and W.G. Macready. No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*, 1(1): 67-82, 1997

C.R. Wren, A. Azarbayejani, T. Darrel, and A. Pentland. Pfinder: Real-Time Tracking of the Human Body, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7): 780-785, 1997

P.N. Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, 311-321, 1993

Genetic algorithm for optimizing fuzzy image pattern matching

Janne Koljonen and Jarmo T. Alander
Department of Electrical Engineering and Automation, University of Vaasa
P.O. Box 700, FIN-65101, Vaasa, Finland
jako@uwasa.fi, jal@uwasa.fi

Abstract

Fuzzy image pattern matching recognizes and localizes objects in images using methods of fuzzy logic. The methods use fuzzy segmentation, templates and operators that bring a heap of alternatives into the reasoning process. Fuzzy pattern matching takes into account uncertainties and inaccuracies of the input images as well as the template images and incorporates them in the matching process. In this paper, a method for fuzzy pattern matching and a method to optimize the matching scheme using a genetic algorithm are proposed. Optimization aims at finding reliable features from a set of calibration images by simultaneously optimizing the segmentation function and the template. Optimization proved to result in good generalization of the matcher for the unseen test images and better performance than the common correlation pattern matching method.

Keywords: computer vision, fuzzy pattern matching, genetic algorithms, object recognition.

1 Introduction

Pattern matching (Sonka *et al*, 1999) is used to recognize and localize objects in images. Other computationally more efficient methods exists for object recognition, e.g. invariant moments (Schallkoff, 1992) and Gabor filters (Kyrki, 2002; Kämäräinen, 2003), but for object localization pattern matching is a competitive method. Pattern matching computes similarities between images and template images or prototypes, which are subjected to transforms like translation and rotation.

Fuzzy logic (Zadeh, 1988) includes uncertainties in logical reasoning. It has been applied to image processing in many ways. Segmentation aims at dividing pixels into similar region i.e. crisp sets. Fuzzy segmentation in turn divides pixels into fuzzy sets i.e. each pixel may belong *partly* to many sets and regions of image (Russakoff *et al*, 2002; Rezaee *et al*, 2000; Tolias *et al*, 1998; Liu *et al*, 1994).

Fuzzy logic has been applied to image enhancement (Choi *et al*, 1997) and edge detection (Russo, 1998) to improve robustness against noise. Fuzzy logic also provides a means to linguistically express image processing operators. Wu *et al*. (1999) have developed a face detection algorithm that utilizes fuzzy pattern matching based on two-term fuzzy similarity relations. Fuzzy morphology is in turn an example how binary image processing operators have been fuzzyfied (Gader, 1997).

Genetic algorithms (GA) have been applied to practically every step of image processing and pattern recognition. References of GA used in fuzzy systems and pattern recognition can be found e.g. in the bibliographies of Alander (1995; 2002), respectively. Tao *et al*. (2003) have optimized membership functions to attain three-level segmentation with maximum fuzzy entropy using GA. Han *et al*. (2001) used GA for stereo vision matching. They took the advantage of GAs in multi-criteria optimization and optimized simultaneously similarity matching and disparity smoothness.

Yokoo *et al*. (1996) developed an algorithm to detect faces based on edges and ellipse matching: Binary edge map is smoothed to a multi-level image and the GA generated ellipses are matched using pixel-wise similarity metric.

In this paper, a novel approach for fuzzy pattern matching and optimization by genetic algorithms is introduced. In section 2, the fuzzy pattern matching scheme is introduced. A cost function to globally optimize the matcher is proposed in section 3. In sections 4 and 5, the method is tested and compared with the common pattern matching method based on Pearson correlation coefficient.

2 Fuzzy pattern matching

In pattern matching, as used in this paper, each class is represented by a prototype vector of features and

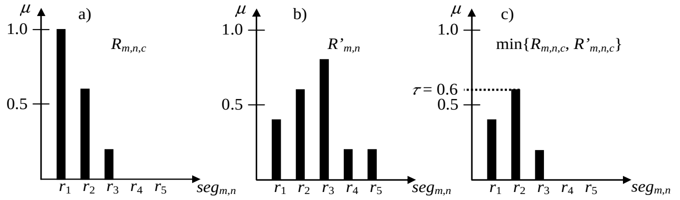


Figure 1: a) *A priori* possibility of segmentation, b) obtained fuzzy segmentation for pixel (*m*,*n*), c) approximate reasoning using sup-min composition (Passino *et al*, 1998) to solve the degree of truth that $seg_{m,n}$ is $R_{m,n,c}$.

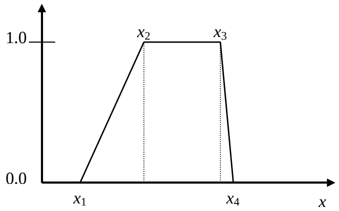


Figure 2: Four turning points of a trapezoidal membership function used for fuzzy segmentation.

a similarity measure is used to compare given patterns to the prototypes. In image analysis the prototype vector is typically a gray-scale template image and correlation is the similarity metric:

$$\gamma = \frac{c_{fw}}{s_f s_w}, \tag{1}$$

where *c* is covariance of image *f* and template *w*, and *s* are the respective standard deviations (Gonzalez *et al*, 2002).

Other common prototype vectors are binary template images that are matched to segmented images. Also e.g. minimum distance (Sonka *et al*, 1999) and Hausdorf distance (Dubuisson *et al*, 1994; Rucklidge, 1996) metrics are used.

For multi-class object recognition pattern matching is computationally inefficient if all transformation combinations are tested, but for object localization pattern matching is feasible. Particularly, if the search space can be restricted to small intervals of only a few free parameters, computational cost is reduced significantly.

Fuzzy pattern matching (FPM) is a modification of classical pattern matching. It is a class of different approaches that are based on fuzzy similarity relations and fuzzy reasoning. FPM can operate either on raw color/gray-scale images or fuzzy segmented images. FPM is supposed to be more robust against noise and distortions than classical pattern matching. Moreover, fuzzy methods resemble human reasoning and results often in intuitively understandable linguistic rules.

The FPM strategy used in this paper relies on a given fuzzy segmentation and fuzzy reasoning. The objective of fuzzy segmentation is to divide the input image into a number of homogeneous regions presented as fuzzy sets. Hence, each pixel can be partially a member of several regions with different membership grades. From another point of view, pixels form fuzzy sets, where the segmentation regions are objects. Fuzzy segmentation methods are e.g. *fuzzy thresholding*, fuzzy clustering (Rezaee *et al*, 2000; Tolias *et al*, 1998; Dunn, 1973, 1974) and fuzzy reasoning that can subsequently be based on color, borders, texture, etc.

Fuzzy pattern matching operating on fuzzy segmented images is based on fuzzy reasoning. It assumes that a spatial pattern of segmented regions implies a target object class. In a simple form, the template is a map of the desired segmented fuzzy sets. Thus the template forms a fuzzy rule base, where each pixel has a separate rule that imply the target class. The firing strength of each rule, i.e. the truth of the antecedent part, is the membership grade of the pixel in question in the desired segmented fuzzy set. The overall truth is aggregated over all rules. A single rule is as follows:

$$\textbf{IF } seg_{m,n} \textbf{ is } R_{m,n,c} \textbf{ THEN } class \textbf{ is } C_{m,n,c}, \tag{2}$$

where $seg_{m,n}$ is the fuzzy set of segmentation at pixel (*m*, *n*) of ROI and $R_{m,n,c}$ is a MF describing *a priori* possibilities of segmentation. Because both *seg* and *R* are fuzzy sets, the strength of firing is solved by approximate reasoning that computes the degree of match between the given and desired MFs. Figure 1 shows how approximate reasoning is applied for a five-class fuzzy segmentation matching.

In more complicated fuzzy pattern matching schemes, the template can include more information. Each pixel may have e.g. a different weight corresponding to the reliability of the pixel. Moreover, the fuzzy connectives, e.g. implication, may be pixel-dependent.

3 Genetic algorithm

Modeling the problem thoroughly requires effort when applying genetic algorithms to any domain. Modeling includes the selection of free model parameters, their genetic coding, and the formulation of an objective function to be optimized.

3.1 Trial coding

The fuzzy pattern matching method utilized in this study consists only of color based fuzzy segmentation and fuzzy implication because it turned out that this quite simple model is complex enough to solve the problem. Moreover, the FPM system is optimized to one target class at a time, as only one object is to be localized, and consequently there is only one global segmentation rule and one template to be coded and optimized.

The simpler the model is the faster and more reliable the optimization. This justifies the void of excess free parameters such as conjunction method in the GA model. That is why some parameters can be chosen arbitrarily as long as the results are satisfactory.

3.1.1 Segmentation

Segmentation is done on the basis of local HSB (Hue-Saturation-Brightness) values. A global fuzzy threshold rule is used to segment the input image, which may be pre-processed, into two complementary fuzzy regions. The regions may be regarded as foreground (FG) and background (BG) even though the matching scheme does not know such differentiation, and in fact sometimes the intuitive classification might name the regions *vice versa*. The segmentation rule for foreground is:

$$\begin{array}{l}\textbf{IF } h_{m,n} \textbf{ is } H \text{ and } s_{m,n} \textbf{ is } S \text{ and } b_{m,n} \textbf{ is } B \\ \textbf{THEN } seg_{m,n} \textbf{ is } FG\end{array}, \quad (3)$$

where $h_{m,n}$, $s_{m,n}$, and $b_{m,n}$ are the hue, saturation, and brightness, respectively, of pixel (m, n) of the input image. H, S, and B are the membership functions of the desired foreground segmentation. For the AND operator Reichenbach's conjunction (multiplication) was used. Each pixel of the fuzzy conclusion variable $seg_{m,n}$ can have in this case two values: *fg* or *bg*. The consequence fuzzy set *FG* is typically a crisp set: *FG* = {(*fg*, 1.0), (*bg*, 0.0)}. The whole fuzzy inference process is illustrated in Figure 3.

Considering one value of $seg_{m,n}$ enables to extract fuzzy sets for foreground and background. E.g. the fuzzy set describing the segmented region for the foreground is obtained by collecting the membership grades of *seg* = *fg* into another fuzzy set: $seg_{r=fg}$ = {((0,0), $\mu_{fg}(seg_{0,0})$),…, ((m,n), $\mu_{fg}(seg_{m,n})$),…}, where $\mu_{fg}(seg_{m,n})$ refers to the membership grade. The segmented fuzzy sets form a 2D lattice that can be visualized by rescaling intensity values. Because the segmented regions are in this case complementary, the membership grades for background are obtained by negation:

$$\mu_{bg}\left(seg_{m,n}\right) = 1 - \mu_{fg}\left(seg_{m,n}\right). \quad (4)$$

Coding the segmentation rule requires modeling the membership function H, S, and B. The selection of trapezoidal MFs leads to four free parameters per MF (Figure 2), provided the low and high levels are fixed to 0 and 1, respectively.

The four turning points must obey the inequality $x_1 \le x_2 \le x_3 \le x_4$. Therefore relative coding of the points is advisable to avoid illegal trials. The second turning point x_2 was selected as the absolute point. The rest are as follows:

$$x_1 = x_2 - \Delta_1,\ x_3 = x_2 + \Delta_2,\ x_4 = x_2 + \Delta_2 + \Delta_3. \quad (5)$$

Consequently, the four parameters to be coded genetically are x_2, Δ_1, Δ_2, and Δ_3. Four bits were reserved for each of these parameters and henceforth they have $2^4 = 16$ different values that were scaled to the range of the input parameters hue, saturation, and brightness. Consider, for instance, that the genotype representation for x_2 = 0110. The normalized phenotype would be 6/15 = 0.4. One drawback of the MF modeling scheme is that the x_i parameters have varying interrelationships. Specifically, e.g. x_4 depends on three free parameters (x_2, Δ_2, and Δ_3) whereas x_2 on only one.

3.1.2 Template

To avoid free parameters template coding is kept as simple as possible. Hence each pixel of the template has a single free parameter that determines the desired segmented region of the two alternatives or that the pixel is omitted altogether. Consequently, each pixel is modeled by two bits. Two bit combinations are reserved for omission and one for each segmented fuzzy set.

What is actually modeled is the membership functions $R_{m,n,c}$ in eq. (2), $C_{m,n,c}$ can be fixed to 1 for the only target class (*class* = 1), and 0 for the opposite class (*class* = unknown) to be formal. Now the alternatives for $R_{m,n,1}$ are fixed to *FG* = {(*fg*, 1.0), (*bg*, 0.0)} and *BG* = {(*fg*, 0.0), (*bg*, 1.0)}. The omission of a whole rule cannot be modeled by any MF without having effect on the conclusion. Hence that case is dealt separately in aggregation step. The resulting fuzzy matching rule alternatives are:

$$\begin{array}{l}\textbf{IF } seg_{m,n} \textbf{ is } FG \textbf{ THEN } class \textbf{ is } C \text{ and} \\ \textbf{IF } seg_{m,n} \textbf{ is } BG \textbf{ THEN } class \textbf{ is } C\end{array}. \quad (6)$$

The overall fuzzy decision making system is illustrated in Figure 3 in terms of one picture element (m,n). The pixel is expected to belong to the background. In this case its membership grade on the fuzzy set background is 0.4 and consequently the support of the pixel for the target class classification is somewhat low. However, the overall conclusion for a ROI is aggregated over all pixels of ROI.

Figure 4 depicts an example of a small template. It is noteworthy that the Hamming distance from *FG* to *omit* and from *BG* to *omit* is 1, which enables efficient optimization by genetic operators, both by crossbreeding and mutation.

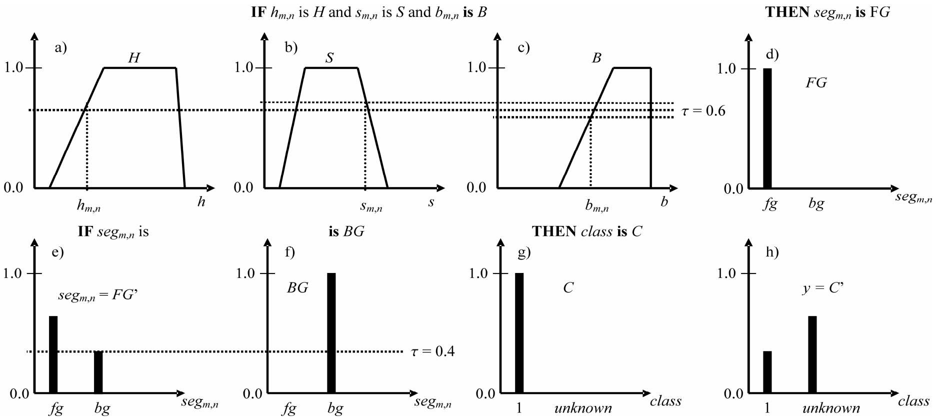


Figure 3: Schematic of FPM inference method. a-c) The antecedent part and d) the conclusive part of the segmentation rule. e) Result of segmentation and premise for the classification rule. f) Antecedent part and g) conclusive part of the classification rule. h) Result of classification.

BG	*BG*	*BG*
omit	*omit*	*omit*
FG	*FG*	*FG*

01	01	01
00	11	11
10	10	10

Figure 4: A 3×3 template: MFs and binary coding.

3.2 Cost function

The objective function (cost function, fitness function) is a scalar evaluation result of a trial. Genetic algorithm is driven to an optimum by the aid of the fitness function. In a multi-criteria optimization task, the different objectives are combined to a single scalar, but the structure of the cost function has a significant effect on the optimization speed and robustness i.e. the cost function has to guide the GA efficiently to the optimum (or one of the optima).

The objectives of fuzzy pattern matching optimization are accuracy and robustness, which mean good localization and discrimination of incorrect positions, respectively. Accuracy is measured by Euclidian distance from the best match to the reference point (m_0, n_0) that is regarded as the correct location of the target object:

$$dist(y) = \sqrt{(m-m_0)^2 + (n-n_0)^2}, \max_{m,n}\{\mu(y_{m,n})\}, \quad (7)$$

where y is the 2D conclusion fuzzy set.

Optimizing discrimination means maximizing the degree of match at the reference point and minimizing it elsewhere. However, the degree of match has usually a strong spatial correlation in images. Hence the minimization criterion is taken outside a suitable distance from the reference point. A *discrimination factor* (*DF*) is now defined by the maximum overall match (best match) and the maximum match outside a given distance r:

$$DF(y) = \max_{m,n}\{\mu(y_{m,n})\} - \max_{\substack{m,n \\ \|(m,n)-(m_0,n_0)\| \geq r}}\{\mu(y_{m,n})\}. \quad (8)$$

Discrimination factor is independent of the mean level of match. If the level of best match needs to be included in the cost function, the following weighted *DF* can be used:

$$DF(y;a) = a\max_{m,n}\{\mu(y_{m,n})\} - (2-a)\max_{\substack{m,n \\ \|(m,n)-(m_0,n_0)\| \geq r}}\{\mu(y_{m,n})\}. (9)$$

Selecting a = 1 eq. (9) reduces to eq. (8). A more general way to include both *DF* and the maximum degree of match to a combined criterion is to use fuzzy logic and connectives. One can for example express linguistically that the objective is to maximize both the maximum match and *DF* (or alternatively either). This *certainty factor* (*CF*) is:

$$CF(y) = \max_{m,n}\{\mu(y_{m,n})\} * DF(y), \quad (10)$$

where ∗ is a fuzzy connective.

The overall fitness function is a combination of a distance term and the modified *DF*:

$$\varepsilon(y) = \begin{cases} c_0 - c_2 \cdot DF(y;a), & dist(y) < d_1 \\ c_0 + c_1 \cdot dist(y)/2 - c_2 \cdot DF(y;a), & d_1 \leq dist(y) < d \\ c_0 + c_1 \cdot dist(y) - c_2 \cdot DF(y;a), & \text{otherwise.} \end{cases} (11)$$

where c_i and d_i are appropriate constants. Bias c_0 adjusts the minimum of ε to 0. If a = 1, c_0 = c_2 because max(DF(y;1)) = 1 and min($dist$(y)) = 0.

The cost function is to be minimized, because as the distance decreases and the *DF* increases, the cost function decreases. The effect of distance is reduced gradually near the reference point. Distances under d_1 are omitted in the fitness function, because they are considered good enough and because the determination of the reference point also has an error of a few pixels. The idea is to first guide the distance close to optimum, after which emphasis is on the *DF*.

4 Tests: nose and iris

Human nose and iris localization was selected for the test case, because the problem is intuitively challenging and it can be applied in the

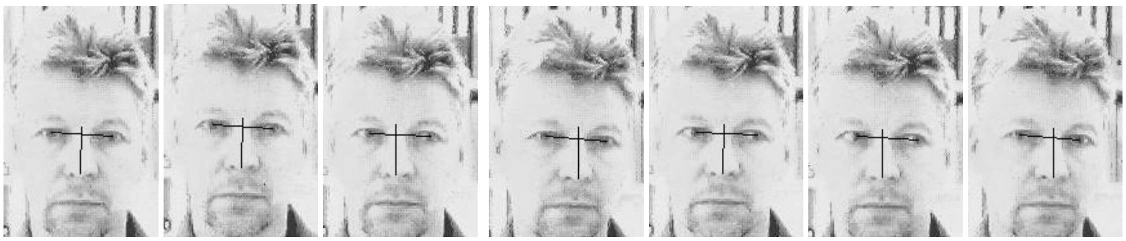


Figure 5: Determination of reference points for calibration and test images (in arbitrary order).

determination of the gaze direction, which in turn has many applications e.g. in human-machine interaction. Measuring the distances from the irises to the centerline of the nose and the distance between irises solves the gaze direction relative to the skull, but it needs calibration. The orientation of the skull could be solved from some rotation variant geometrics. The movements of the eyes are small, the centerline of the nose is difficult to find, and the images are obtained by a low resolution webcam that make the task even more difficult.

The test images have been taken from almost the same angle, distance, rotation, and lighting conditions, which makes it possible to concentrate to vary only translation when matching, and thus to reduce the complexity of computations. To make the algorithm simpler and faster the face localization step was made manually, even though numerous algorithms for that purpose have been reported in the literature. Thus the ROI for nose localization was the limited approximately to the area of the face, consisting of 120×180 pixels.

In a fully automated application, the search would be hierarchical e.g. by locating subsequently face, nose, right eye, and left eye, and using *a priori* knowledge at each step. However, now the objective is to study fuzzy pattern matching and optimization.

Three reference images with different direction of sight were included in the calibration set and four in the test set. The determination of reference points of nose are demonstrated in Figure 5.

In all test runs, the size of the template was: for nose 42×52 and for iris 10×6 pixels. In order to reduce the number of free parameters the template for nose recognition consisted of 3×3 blocks of equal membership function. Consequently, the nose template had 14×18 = 252 free parameters and the iris templates 60. Because segmentation required 4 MFs, each 4 parameters of 4 bits, and each template parameter required 2 bits, there were totally 4×4×4+252×2 = 268 bits (ca. $4.7\cdot10^{80}$ alternatives) for nose localization and 64+60×2 = 184 bits for iris localization to be optimized.

The cost function in eq. (11) is defined for one FPM result. For three calibration images the cost function was the maximum of the three cost function evaluation ε_i. The result corresponds to the worst case.

$$\varepsilon_{tot}(y)=\max\{\varepsilon_1(y),\varepsilon_2(y),\varepsilon_3(y)\}. \tag{12}$$

However, the cost function of the following calibration image was evaluated only if the temporary fitness result was below 1000 and better than the worst fitness in the elite. This reduced the complexity of computation at the early stages.

Genetic operators were: uniform crossover, one-point crossover, and bitwise mutation. Population size was rather small (50) and 20 offspring were created each generation by roulette wheel selection from the elite. The small population size corresponds to quite local search strategy, which can be fast but, on the other hand, unreliable. Balancing between speed and reliability is always a compromise. After a few test runs, the cost function parameters were fixed to: a = 1, c_0 = 1000, c_1 = 50, c_2 = 1000, d_1 = 2, d_2 = 2. Radius r in *DF* was half of the edge of the template.

The traditional correlation pattern matching (eq. 1) was used as a reference method. A template of size equal to the FPM template was created by cropping the RGB channels around the target location from one image and the rest of the images were used as test images.

5 Results

This section reports the experimental results: accuracy and robustness measures of the optimized FPM system and statistics of the GA optimization. The results are discussed particularly as for the generalization ability of the FPM system.

5.1 Nose localization

Two GA runs of 500 generations were carried out in order to optimize the nose localization fuzzy pattern matching system. In run 1, raw images were used, but in run 2 the images were preprocessed by color preserving equalization.

The results for three calibration images and four test images are summarized in Table 1. The results indicate that the optimized matcher generalizes well to the unseen test images whose pose, scale and

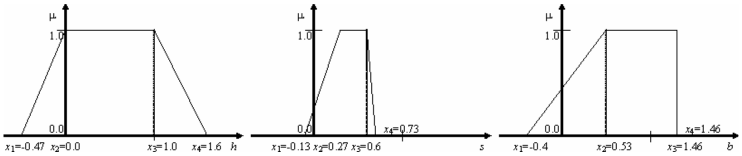


Figure 6: Membership functions of run 2 for nose matching.

lighting conditions are nearly equal, but where the pose of the eyes varies. The generalization ability is probably based on the simultaneous feature creation (segmentation) and feature selection (template optimization). The GA selects pixels with coherent information in the calibration images. Those pixels are potentially inter-image-invariant features that are present in the test images, too.

The reference method (correlation) also managed to generalize to the test images. The location error was smaller but the discrimination factor was poorer, which may cause severe miss location under a noisier environment.

Figure 6 illustrates the optimized membership functions of run 2. Hue has evidently no role in segmentation as it is constant 1 in [0, 1], the range of normalized hue values. Similarly, the influence of brightness is marginal as in nose and its surrounding brightness values under 0.53 are rare. As a conclusion, saturation is the main feature that contributes on segmentation in this case. Specifically, pixels around the nose have usually $s > 0.27$ and pixels on the nose $0.18 \leq s \leq 0.27$.

Figure 7 shows the optimized template, matched image, segmentation result, and the map of degrees of match. The spatial pattern of the optimal template is such that contributing pixels are mostly located on the borders of the nose where contrast is high. The most important regions are the corners of the eyes, nostrils, and the upper jaw. In contrast, the side slopes of the nose are more or less unreliable features due to poor contrast and sensitivity to lighting conditions. Figure 7 d) visualizes discrimination and spatial autocorrelation of FPM. The degree of match around the maximum has a circular correlation pattern. However, outside the neighborhood the degree of match is systematically much lower indicating good discrimination.

Optimizing FPM for nose localization required totally 10,000 trial evaluations. Figure 8 shows the development of the fitness value taken from run 1. Figure 9 shows the development of the components of the cost function: location error and discrimination factor. Statistics indicate that after a few dozens of generations the location error has settled within the tolerance margin of one or two pixels. After finding the correct location, the discrimination factor starts to grow as planned.

Table 1: Nose matching after 500 generations. *Dist* is location error in pixels, *max* the maximum degree of match, and *DF* is according to eq. (8). 'reference' is the benchmark method in eq. (1).

	run 1			**run 2**			**reference**		
image	*dist*	*max*	*DF*	*dist*	*max*	*DF*	*dist*	*max*	*DF*
cal1	2	0.94	0.30	1	0.86	0.26	0	1.00	0.19
cal2	2	0.94	0.30	0	0.86	0.26	1	0.96	0.17
cal3	1	0.95	0.32	1	0.86	0.26	1	0.97	0.18
test1	2	0.92	0.28	2	0.87	0.25	1	0.92	0.18
test2	3	0.93	0.29	2	0.85	0.25	1	0.97	0.17
test3	4	0.90	0.21	0	0.86	0.20	1	0.96	0.17
test4	4	0.87	0.21	2	0.84	0.19	1	0.96	0.17

The range of fitness values in the elite (Figure 8) can be used to estimate the maturity of the population. As the difference approaches zero it becomes unlikely to have any further improvement.

5.2 Iris localization

In addition to nose, the irises of the eyes were located using optimized FPM. Here the results for the right eye (from the readers' point of view) are given.

The same nine images as in nose localization were used, but the calibration images were selected so that they included different poses of eyes. The results of two GA runs of 150 generations (Table 2) prove again good generalization, but the *DF*s are lower than in nose localization. The explanation might be the smaller size of the template. Thus it is more probable to randomly have better matches

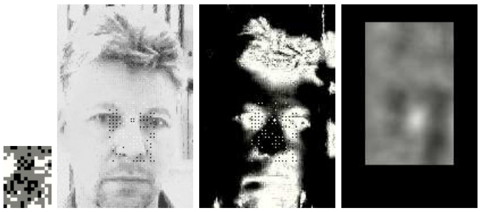


Figure 7: Run 2 results (equal scale). a) Optimized template (gray = don't care), b) Template overlaid the best match position on original (■=origin), and c) fuzzy segmented image (white cross=origin), d) Degree of match (totally black area = uncomputed).

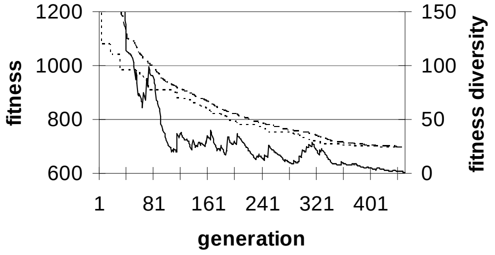


Figure 8: Evolvement of best (dotted line) and worst fitness (dashed) of the elite in nose run 1. solid line = fitness diversity. (Ordinates cut to magnify the relevant part).

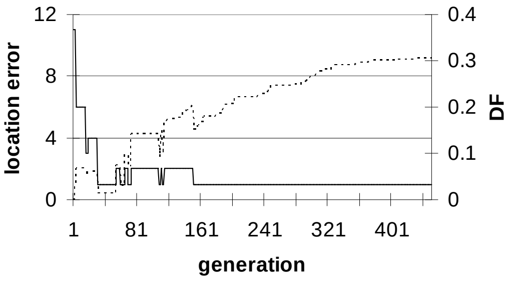


Figure 9: Evolvement of location error (solid) and discrimination factor (dotted line).

outside the best match.

Now the reference method failed in 5 of 6 of its test images. The negative *DF* was computed by subtracting the maximum degree of match from degree of match at the desired location. This example clearly demonstrates the justification of optimizing also the discrimination factor.

Analyzing the MFs in Figure 10 and the HSB values of the images implies that the brightness value is now the dominant feature. In fact, the turning point b = 0.6 seems to be a good selection to separate iris and pupil (b < 0.6) from the surroundings (b > 0.6). The crucial point is to recognize the white of the eye, which is the varying feature as the eye turns.

The segmentation results shown in Figure 11 are similar even though the segmentation membership functions are different. This validates the conclusion that brightness was the dominant feature as its membership functions are the only similar ones. Segmentation results also demonstrate that the white of the eye are separated from the iris. Presumably the selection calibration images with different eye poses forced the selection of such membership functions.

6 Conclusion

It was shown how fuzzy logic is used to fuzzy segmentation and pattern matching. Fuzzy pattern matching incorporates uncertainties inherent in measurement data like image, incorporates it in computational steps and results. The output is a two dimensional fuzzy set that describes the possibilities of different locations of the target object.

Table 2: Results for iris after 150 generations.

	run 1			run 2			reference		
image	*dist*	*max*	*DF*	*dist*	*max*	*DF*	*dist*	*max*	*DF*
cal1	2	0.95	0.12	2	0.92	0.15	0	1.00	0.05
cal2	2	0.91	0.12	1	0.95	0.18	1	0.96	0.02
cal3	2	0.96	0.13	2	0.94	0.16	72	0.79	-0.14
test1	2	0.90	0.08	3	0.95	0.16	85	0.87	-0.08
test2	4	0.96	0.15	2	0.97	0.17	46	0.89	-0.06
test3	3	0.96	0.11	2	0.94	0.16	70	0.92	-0.02
test4	3	0.96	0.09	2	0.90	0.06	65	0.89	-0.05

The fuzzy pattern matching method used in this study consisted of three membership functions in HSB color space to perform a fuzzy segmentation. The template included the desired spatial pattern of the segmented pixels. Both the membership functions and the template were optimized using a stochastic search strategy, genetic algorithm. The results show that the optimized fuzzy pattern matching system generalizes well to the test images. However, the set of images was small and did not include much noise. Nevertheless, the novel method outperformed the traditional correlation pattern matching drastically in many cases.

Acknowledgements

The Research Institute for Technology at the University of Vaasa supported financially J. Koljonen. Timo Rautakoura and Hannu K. Niinimäki are acknowledged for the test case images and the anonymous reviewers for their precious comments and suggestions.

References

J.T. Alander. An Indexed Bibliography of Genetic Algorithms in Pattern Recognition. *Report* 94-1-PATTERN, University of Vaasa, 2001. URL: ftp://ftp.uwasa.fi/cs/report94-1/.

J.T. Alander. An Indexed Bibliography of Genetic Algorithms with Fuzzy Systems. *Report* 94-1-FUZZY, University of Vaasa, 1995. URL: ftp://ftp.uwasa.fi/cs/report94-1/.

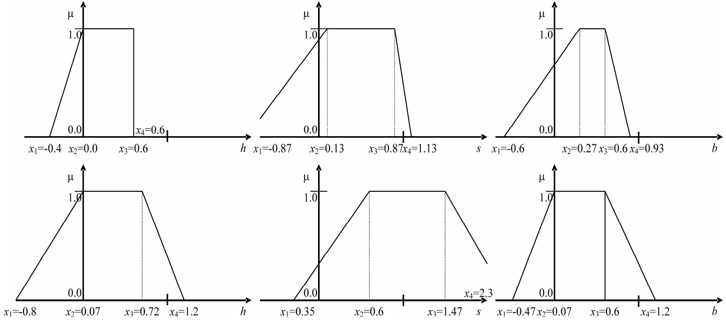


Figure 10: MFs for run 1 (upper) and run 2 (lower) for iris matching.

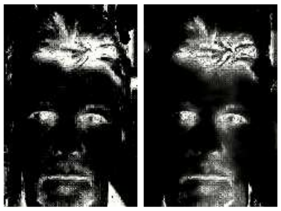


Figure 11: Fuzzy segmented images of run 1 (left) and run 2 (right) for iris localization.

Y.S. Choi, R. Krishnapuram. A Robust Approach to Image Enhancement Based on Fuzzy Logic. *IEEE Transactions on Image Processing*, **6**(6), 1997.

M-P. Dubuisson, A.K. Jain. A modified Hausdorff distance for object matching. In: *Proceedings of the 12th IAPR Int. Conf. on Pattern Recognition*, **1**: 566–568, 1994.

J.C. Dunn. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, **3**: 32–57, 1973.

J.C. Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*, **4**: 95–104, 1974.

P.D. Gader. Fuzzy Spatial Relations Based on Fuzzy Morphology. IEEE, 1997.

R.C. Gonzalez, R.E. Woods. *Digital Image Processing*. Second edition. Prentice-Hall, New Jersey, 2002.

K-P. Han, K-W. Song, E-Y. Chung, S-J. Cho, Y-H. Ha. Stereo matching using genetic algorithm with adaptive chromosomes. *Pattern Recognition*, **34**: 1729–1740, 2001.

V. Kyrki. Local and Global Feature Extraction for Invariant Object Recognition. *Acta Universitatis Lappeenrantaensis* 133, 2002. PhD thesis.

J-K. Kämäräinen. Feature Extraction Using Gabor Filters. *Acta Universitatis Lappeenrantaensis* 165, 2003. PhD thesis.

J. Liu, Y-H. Yang. Multiresolution Color Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **16**(7), 1994.

K.M. Passino, S. Yurkovich. Fuzzy Control, Addison-Wesley, California, 1998.

M.R. Rezaee, P.M.J. van der Zwet, B.P.F. Lelieveldt, R.J. van der Geest, J.H.C. Reiber. A Multiresolution Image Segmentation Technique Based on Pyramidal Segmentation and Fuzzy Clustering. *IEEE Transactions on Image Processing*, 9(7), 2000.

W. Rucklidge. Efficient visual recognition using the Hausdorff distance. In: *Lecture Notes in Computer Science*, **1173**, 1996.

D.B. Russakoff, T. Rohlfing, C.R. Maurer Jr. Fuzzy segmentation of X-ray fluoroscopy image. *Medical Imaging* 2002: *Image Processing*. *Proceedings of SPIE* **2684**, 2002.

F. Russo. Edge Detection in Noisy Images Using Fuzzy Reasoning. *IEEE Transactions on Instrumentation and Measurement*, **47**(5), 1998.

R. Schallkoff. Pattern Recognition – Statistical, structural and neural approaches, John Wiley & Sons, Inc., New York, 1992.

M. Sonka, V. Hlavac, R. Boyle. *Image Processing, Analysis, and Machine Vision*. Second edition. Brooks/Cole Publishing Company, USA, 1999.

W-B. Tao, J-W. Tian, J. Liu. Image segmentation by three-level thresholding based on maximum fuzzy entropy and genetic algorithm. *Pattern Recognition Letters*, **24**: 3069–3078, 2003.

Y.A. Tolias, S.M. Panas. Image Segmentation by a Fuzzy Clustering Algorithm Using Adaptive Spatially Constrained Membership Functions. *IEEE Transactions on Systems, Man, and Cybernetics–Part A: Systems and Humans*, **28**(3), 1998.

Y. Yokoo, M. Hagiwara. Human Faces Detection Method using Genetic Algorithm. In: *Proceeding of IEEE Int. Conf. on Evolutionary Computation*, 113–118, 1996.

H. Wu, Q. Chen, M. Yachida. Face Detection From Color Images Using a Fuzzy Pattern Matching Method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **21**(6), 1999.

L.A. Zadeh. Fuzzy Logic. *IEEE Computer*, **21**(4): 83–93, 1988.

Effects of population size and relative elitism on optimization speed and reliability of genetic algorithms

Janne Koljonen and Jarmo T. Alander
Department of Electrical Engineering and Automation, University of Vaasa
P.O. Box 700, FIN-65101, Vaasa, Finland
jako@uwasa.fi

Abstract

A good selection of the internal parameters of a genetic algorithm may reduce the optimization speed considerably. We studied the effects of two key parameters, namely population size and relative elitism, on the optimization speed and reliability. We found that changing one parameter can be compensated for changing another. By penalizing for the unreliability of the genetic algorithm the optimum combination of the parameters shifted. By using a meta-GA we found relationships between the total population size, relative elitism, and the penalty parameter.

Keywords: genetics algorithms, meta-GA, optimization, GA parameters.

1 Introduction

Genetic algorithm (GA) is the most popular artificial evolutionary algorithm used successfully in difficult optimization and search problems. It consists of a population of trials, a fitness function to map the genotype trials to real numbered phenotypes, and a set of genetic operators to create new trials. GA has several parameters to be selected or tuned, and many alternative implementations for any problem. These include the formalization of the objective function and the genotype-phenotype mapping, selection of genetic operators, selection method, and tuning of parameters such as population size, elitism, mutation rate, etc. (Alander, 1994; Alander, 2002: Darwin, 1998; Koljonen *et al*, 2004, Nordling *et al*, 2004).

The selection of the GA parameters and problem representation has a crucial effect on the GA performance – optimization speed and reliability. In literature, these have been intensively studied both theoretically (Alander, 1992; Alander, 2002; Gao, 2003; Rees *et al*, 1999a; Rees *et al*, 1999b) and empirically (Alander, 1992; Alander, 2002; Rees *et al*, 1999a), yet no final conclusions can be drawn due to the diversity of different kinds genetic algorithms and problem domains.

In this paper, we continue (Alander, 1992; Alander, 2004) to study the effect of the population size and relative elitism on optimization speed and reliability. If tuning GA parameters to the optimal speed, e.g. in mean sense, the probability that the GA becomes stuck without finding a solution within the maximum number of generation may increase. That is why we are also interested in the reliability.

2 Hypotheses and approach

We used an empirical approach to study the effect of population and elite size. The GA was run with two test problems and several parameter combinations. With each configuration the statistics of the GA performance were recorded.

The optimization speed was measured by the number of new individuals created i.e. the number of fitness function calls (trials), inclusive the initial population. The optimization reliability was in turn measured by the fitness of the best individual trial at the end of the optimization (when a solution was found or the maximum number of trials was evaluated). We always minimized and the best fitness was zero if a solution was found and larger than zero otherwise.

2.1 Objective parameters

In our genetic algorithm the population is subdivided into elite and immature population consisting of the latest offspring. The population is sorted by the fitness values of the individuals. The parents of the offspring are randomly drawn from the elite by a uniform probability distribution. Consequently, the individuals in the immature population do not have a chance to be selected as parent; it is merely a place to temporarily place the offspring before sorting.

We studied the effect of the sizes of these subpopulations. The size of the elite is denoted by n_e and the size of the immature population by n_i. The total population size is thus $n_e + n_i$ and relative elitism is defined as $n_e/(n_e + n_i)$. By n_{max} we denote the maximum number of individuals allowed to be created during a GA run.

We presumed that the elite size is more important, because it contains the alleles the offspring inherit. Furthermore, we assumed that by having only a small immature population – i.e. by almost immediately either inserting the offspring into the elite, if fit enough, or rejecting it – we could speed up the GA in some cases. We believed that if good gene combinations were produced, it would be profitable to allow them spreading freely. On the other hand, we understood that it could cause premature convergence if its alleles occupied the population too fast. One can interpret that the smaller n_i the more local the search is.

The other principal parameters were fixed. We used *uniform crossover* (Syswerda, 1989) and mutation rate 0.1 genes mutated per individual.

2.2 Test problems

As a test problem, we decided to use a variation of the popular benchmark problems *one-max*, where the *Hamming distance* to the *all ones* bit string is maximized. In our variation, the target bit string is randomly selected, and the fitness value is the Hamming distance (HD):

$$\text{Hamming}(\mathbf{x},\mathbf{y}) = \sum_{i=0}^{n-1} |\mathbf{x}_i - \mathbf{y}_i| \, , \qquad (1)$$

where $\mathbf{x}$ and $\mathbf{y}$ are bit vectors, $\mathbf{x}_i$ and $\mathbf{y}_i$ the i^{th} elements of the vectors, and n the dimension of the vectors.

This problem is GA-easy because the fitness function directly tells how many bits are correct. The more bits are correct the more probable it is that uniform crossover produces solutions. Hence, the fitness function guides the GA effectively.

The other test fitness function (FF) was square-root distance (SRD) of the phenotype to the target phenotype, rounded down to the nearest integer. By changing the target of the problems, the fitness landscape changes. The subscript indicates the target, e.g. SRD_{127}. One can also select the number of bits and thus change the range of the FF. The SRD fitness function is:

$$fitness = \left\lfloor \sqrt{|phenotype - target|} \right\rfloor \cdot \qquad (2)$$

3 Optimization speed

We wanted to study how the elite size and relative elitism affect GA optimization speed. We used 16 bit Hamming distance fitness function and 10 bit square-root FF. With the HD and the SRD the GA was allowed to create at maximum 400 and 800 individual, inclusive the initial population, respectively. The GA was run 200 times with the same population size parameters (n_e and n_i), and the statistics of the optimization speed and reliability were recorded.

3.1 Results and discussion

The population size ranges were the following: For the Hamming distance $n_e = \{8,16,24,32,40,48\}$ and $n_i = \{1,2,3,\ldots,50\}$, and for the square-root distance $n_e = \{8,16,32,64,128\}$ and $n_i = \{1,2,3,\ldots,100\}$. The results are shown in Figures 1, 2, and 3.

In Figure 1, it is seen how the optimization speed depends on the population size parameters when the test function was the Hamming distance and the square-root FF. For the HD, the valley representing the optimal parameter combinations is close to the

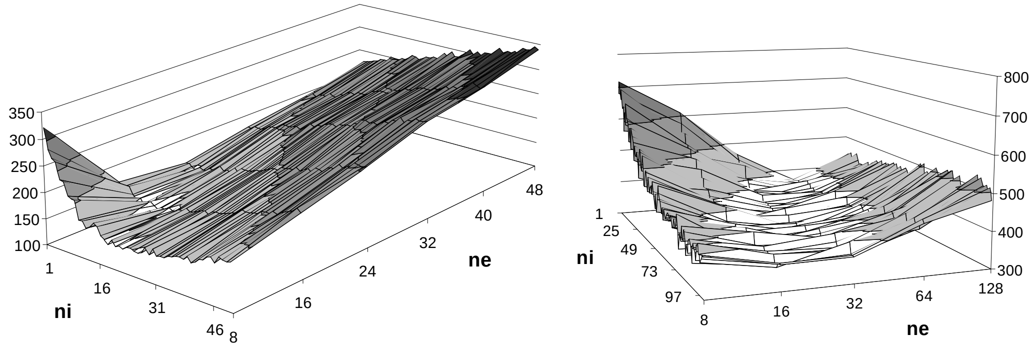


Figure 1: Average number of fitness evaluations with the Hamming distance (left) and the square-root function (right).

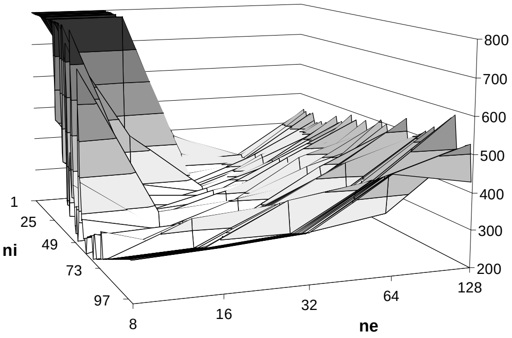


Figure 2: The median value of the optimization speed for SRD is rougher than the average and there is a sharper edge on small population values indicating premature convergence.

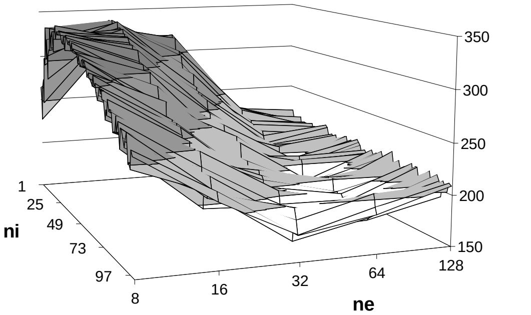


Figure 3: Standard deviation vs. population sizes for SRD.

origin and narrow. Little changes in population size affect optimization speed a lot. Moreover, n_e and n_i seem to be inter-correlated, i.e. decreasing one can be compensated by increasing the other. The SRD problem was more difficult than the HD, over 300 fitness function calls compared to over 100 with the HD, although it had fewer genes than the HD (12 bits *versus* 16 bits). The optimal population size valley is flatter than with the HD.

Results with these problems suggest the opposite to the belief (Alander, 2002) that even smaller than optimal values were not harmful to optimization as the slopes away the optima are relatively equally steep in both directions. The reason for this illusion is probably the optimization speed measure (average of the individual created). With small populations there are both good and bad cases. However, as the maximum number of individuals was limited, the poor cases do not affect the mean value significantly. In this case order statistic measures like median indicate the premature convergence tendency better as Figure 2 demonstrates. At small population sizes the slope of the median is very steep. On the other hand, the graph of the median is rough at the transition points probably due to the statistical properties of median.

An even earlier indication of the risk of premature convergence is given by standard deviation (Figure 3). Actually, sometimes even before mean performance achieves its minimum, standard deviation begins to grow.

4 Optimization reliability

Optimization speed is only half of the story. Particularly in real time GA applications, both speed and reliability are desired. A solution has to be found within a limited time and it has to be good enough. Sometimes sub-optimal solutions are permitted and a compromise between speed and robustness is done.

4.1 Reliability surfaces

We measured the reliability with the best fitness of a GA run for a limited maximum number of fitness function calls. The best fitness was averaged over 200 GA runs. The population sizes were varied in the same manner as mentioned in section 3.1.

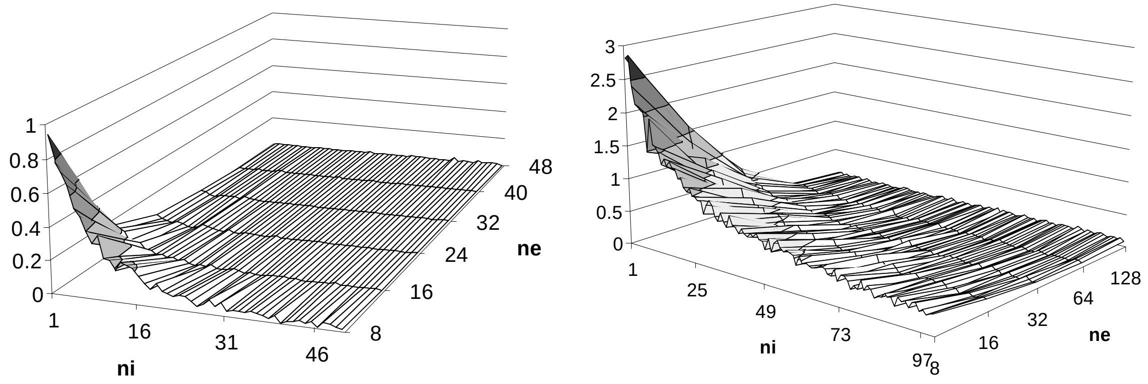


Figure 4: GA optimization reliabilities (mean of the best fitness) of the Hamming distance function (left) and the square-root distance (right) are similar in shape but have different scales.

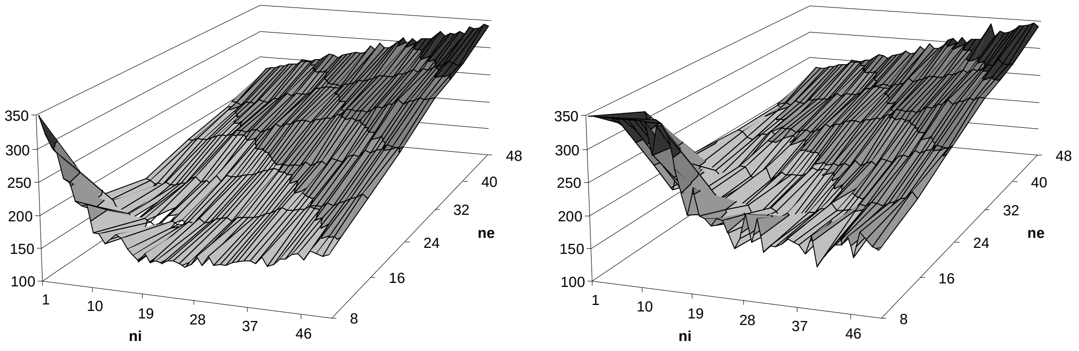


Figure 5: The effect of penalty parameter. The minimum with *a* = 50 (left) is lower and closer to the origin than with *a* = 750 (right). The test fitness function was the Hamming distance.

The optimization reliability surfaces are shown in Figure 4. As presumed, the larger the populations the more reliable the optimization. Full reliability can almost be achieved for both test functions.

However, with speed-optimal population sizes the reliability is not complete. Henceforth, a compromise may be desirable.

4.2 Cost function of speed and reliability

To make a compromise between speed and reliability we need some criterion. Let us define a cost function ε that rewards for speed and penalizes for unreliability. A simple cost function is a linear combination of speed and reliability with a penalty parameter *a*:

$$\varepsilon(a) = fitnessevaluations + a \cdot bestfit\,. \tag{3}$$

4.3 Cost function surfaces

The cost function has only one tunable parameter, whose selection is up to the designer of the GA. The effect of the penalty parameter, i.e. the cost function surface with respect to population sizes, is demonstrated in Figure 5.

The conclusion is that increasing the penalty parameter makes the surface steeper at small population sizes and moves the cost function minimum to larger populations. Because the fitness value is stochastic, it is hard to localize the population size optima with respect to different penalty parameter values. To overcome this difficulty we took the advantage of using another GA, called meta-GA, to find the optima.

5 Meta-GA optimization

The cost function enables to compromise between optimization speed and reliability. If we increase penalty parameter *a*, we emphasize more reliability. We showed in the previous section that changing *a* may move the population size optimum. Next we minimize the cost function with a meta-GA.

5.1 Method

Meta-GA is a method to optimize the GA parameters by another GA. In other words, the problem of selecting parameters is now another optimization problem, which is stochastic due to the stochastic nature of the GA (Mansfield, 1990; Alander, 2002; Bäck *et al*, 2002). In the literature, many variations of meta-optimization have been proposed. Sometimes another GA can be avoided as the GA itself optimizes its parameters (See e.g. Hartono *et al*, 2004).

A meta-GA is similar to a regular GA, but the individuals are GAs, too (Figure 6). The meta-GA runs as many GAs, i.e. the actual optimization problems, as its population size is. The fitness function is

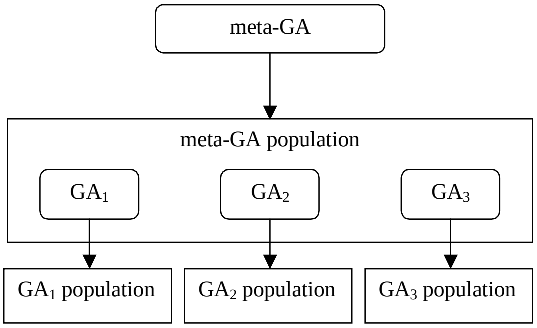


Figure 6: The principle: Meta-GA optimizes the parameters of the actual GAs.

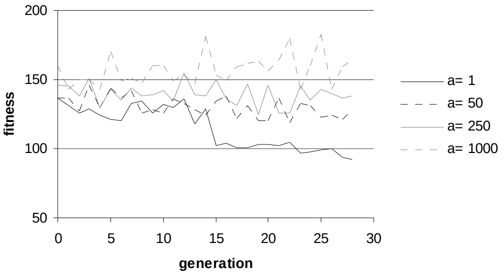


Figure 7: Fitness of the meta-GA ε(*a*) vs. generation of the meta-GA. The stochastic nature of the fitness function is visible.

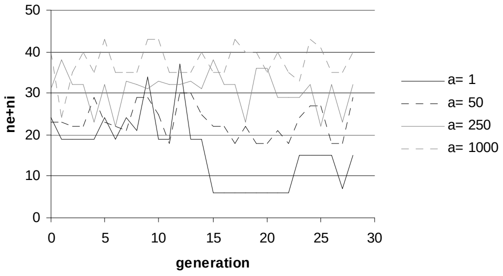


Figure 8: Total population size during optimization. The converged levels with different values of *a* are clearly separated and in the expected order.

defined by measuring GA performance: now we use the cost function ϵ(*a*), which is stochastic.

5.2 Results with Hamming distance

To demonstrate the stochastic nature of the meta-GA fitness function (cost function ϵ) we have plotted the meta-fitness against the meta-GA generation (Figure 7) with four different values of *a*. The n_e and n_i fluctuated heavily during the meta-GA optimization, because the optimum is a flatter valley. Figure 8 shows that the optimal total population size has less fluctuation. During the last ten generations the decrease in fitness was small. Hence we decided to use the average of the last ten meta-GA generations when determining the optimal population sizes with respect the penalty parameter *a*.

The optimal elite and immature population sizes are plotted against *a* in Figure 9. The optimal elite size seems to increase linearly and the immature population size perhaps more or less logarithmically with increasing *a*. However, these conclusions may be too daring considering the small sample size and the internal variation.

The optimal total population size and relative elitism vs. *a* are shown in Figure 10. It indicates quite strong logarithmic dependence (R^2 = 94 %).

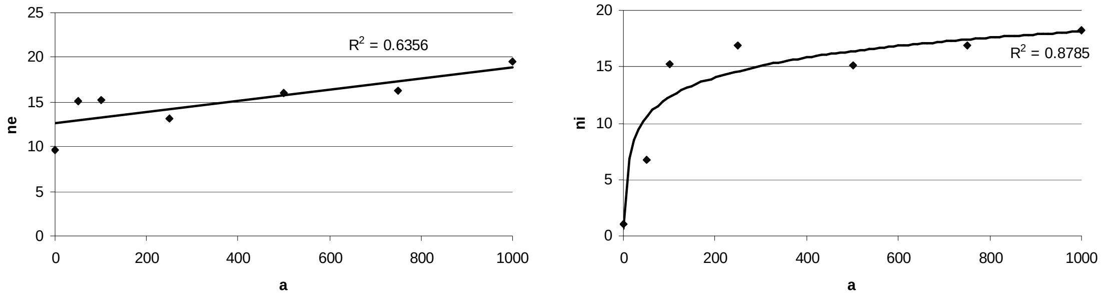


Figure 9: The optimal n_e (left) and n_i vs. penalty parameter. The relationships are quite linear and logarithmic, respectively.

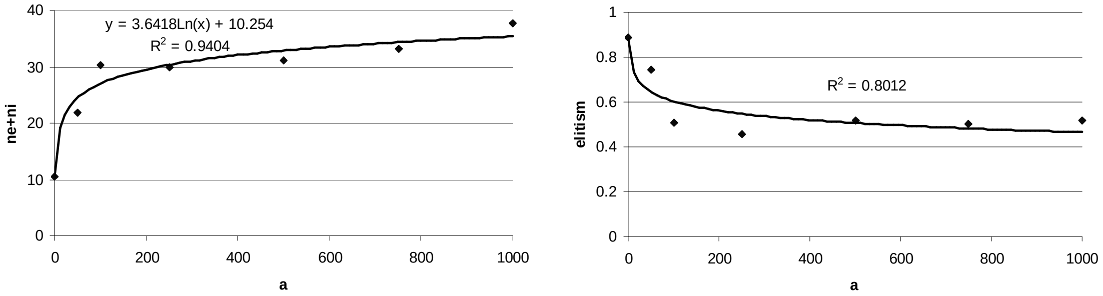


Figure 10: The optimal total population size (left) depends logarithmically on the penalty parameter. Relative elitism vs. the penalty parameter (right).

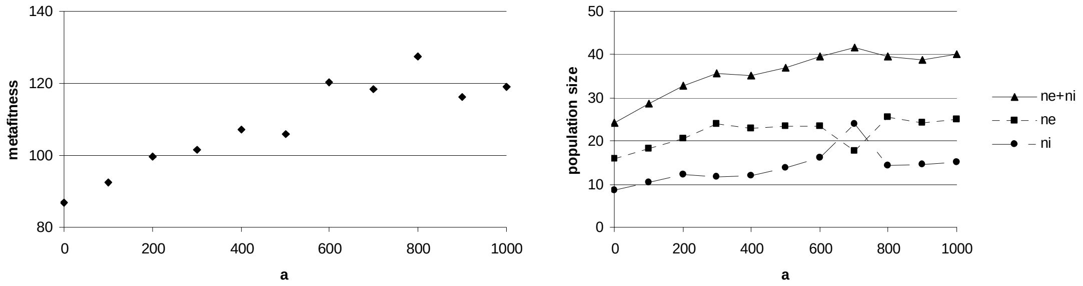


Figure 11: With the easiest square-root function (SRD_0) the penalty factor has an almost linear effect on the meta-fitness (left). Optimal population sizes with respect to *a* with the SRD_0 (right).

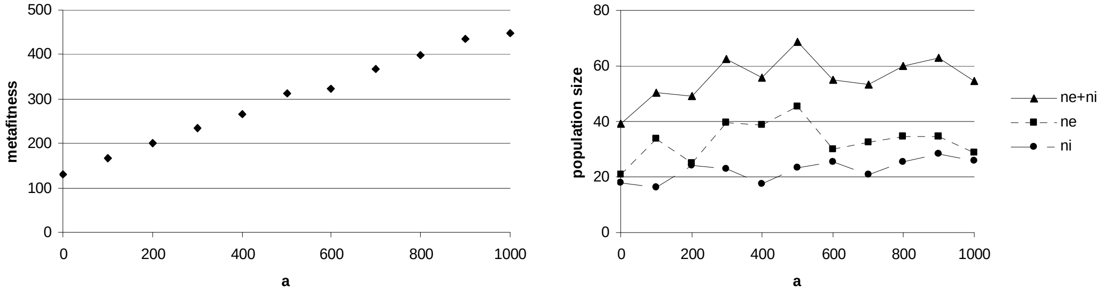


Figure 12: Meta-fitness vs. penalty factor with SRD_{127} (left). Optimal population sizes with SRD_{127} (right).

This result is more reliable as the total population size was found to vary less during optimization (see Figure 10). Yet only qualitative conclusions should be drawn: total population size should be increased logarithmically and elite size almost linearly when increasing *a*. The relative elitism should be in turn decreased, which corresponds to more global search.

5.3 Results with square-root distance

The same procedure of meta-GA optimization was also carried out with the other test problem, square-root distance. Now we used two fixed targets, the GA easiest one (SRD_0) and the GA hardest one (SRD_{127}) of the SRD family, in order eliminate the variation between different targets without the need for impractical number of runs.

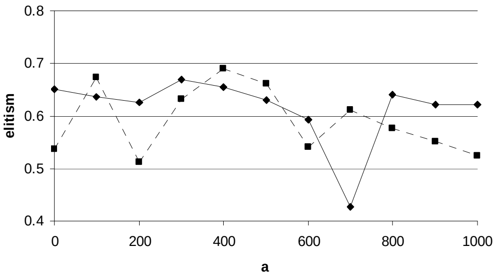


Figure 13: Relative elitism vs. penalty factor with SRD_0 (solid) and SRD_{127} (dotted line)

The meta-fitness depends almost linearly on the penalty parameter with both targets (see Figures 11 and 12). We see also from the meta-fitness that the SRD_{127} is more GA difficult than the SRD_0. We can conclude, as for the SRD_0, that the total population size must be increased quite logarithmically when increasing penalty parameter *a*. The same holds for the population size of the SRD_{127} given in Figure 12. Moreover, we can see that a larger population is required for optimal GA performance for the easy SRD_0 than for the difficult SRD_{127}.

Finally, the relative elitism is plotted against the penalty parameter *a* in Figure 13. Despite high variance we can conclude that the tendency is downward as *a* increases i.e. to a more global search. Furthermore, it suggests higher elitism (more local search) for the easier SRD_0 than for the more difficult SRD_{127}. However, variation is great and no statistically significant conclusion can be drawn.

6 Conclusions

The population size is a key parameter of a genetic algorithm. We have introduced two population size parameters: elite size and relative elitism. We studied their influence on the optimization speed and reliability. The results confirmed the common belief that decreasing population size increases optimization speed to a certain point, after which premature convergence slows the optimization speed down. The optimization reliability in turn usually increases monotonically with increasing population size.

The optimization speed and reliability were combined into a cost function that penalizes for unreliability. The optimal population parameters were searched with a meta-GA approach with a set of penalty parameters. We found that the optimal population size increases and relative elitism decreases (corresponding to more global search) with increasing penalty parameter i.e. higher reliability requirement. However, the results should be considered only qualitatively due to high variance. Moreover, results from the two test functions, Hamming distance and a discrete square-root distance, indicate that the optimization speed and reliability landscapes vs. population size parameters have different shapes. The same holds partly on the optimal population sizes with respect to reliability requirements. As a conclusion, qualitative thumb-of-rules can be given but not precise quantitative ones.

Acknowledgements

Janne Koljonen would like to thank the Research Institute for Technology at the University of Vaasa for the financial support.

References

J.T. Alander. On optimal population size of genetic algorithms. In: *Proceedings of Computer Systems and Software Engineering, 6th Annual European Computer Conference*, 65–70, 1992, IEEE Computer Society, IEEE Computer Society Press.

J.T. Alander. An Indexed Bibliography of Genetic Algorithms: Theory and Comparisons. University of Vaasa, 1994. URL: ftp://ftp.uwasa.fi/cs/report94-1.

J.T. Alander. Potentials of Genetic Algorithms, TEKES, 2002.

J.T. Alander. On the basic genetic algorithm parameters – an empirical study. In: *Proceedings of the 11th Finnish Artificial Intelligence Symposium* (STeP2004), **3:** 19–32, 2004. Finnish Artificial Intelligence Society (FAIS), Vantaa (Finland).

T. Bäck, D.B. Fogel, Z. Michalewicz (eds.). *Evolutionary Computation* 2, 2002, ISBN: 0750306653.

C. Darwin. *On the Origin of Species*. Reprint of the 1st Edition, published Nov. 1859, Wordsworth Limit, 1998, Great Britain. ISBN: 1853267805.

Y. Gao. Population size and sampling complexity in genetic algorithms. In: *Proceedings of the Bird of a Feather Workshops* (GECCO03), 178–181, 2003.

P. Hartono, S. Hashimoto, M. Wahde. Labeled-GA with adaptive mutation rate, In: *Proc. IEEE CEC*, 1851–1858, 2004.

J. Koljonen, J. Lappalainen, J.T. Alander, A. Backman. LEDall – adaptive LED lighting system. In: *Proceedings of the 11th Finnish Artificial Intelligence Symposium* (STeP2004), **3**: 115–125, 2004. Finnish Artificial Intelligence Society (FAIS), Vantaa (Finland).

R.A. Mansfield. *Genetic Algorithms*, University of Wales, 1990.

T.E.M. Nordling, J. Koljonen, J.T. Alander, P. Geladi. Genetic algorithms as a tool for wavelength selection. In: *Proceedings of the 11th Finnish Artificial Intelligence Symposium* (STeP2004), **3**: 99–113, 2004. Finnish Artificial Intelligence Society (FAIS), Vantaa (Finland).

J. Rees, G.J. Koehler. An investigation of GA performance results for different cardinality alphabets. In: *IMA Volumes in Mathematics and its Applications Proceedings from the IMA Workshop on Evolutionary Algorithms*, 191–206, 1999a, Springer, New York.

J. Rees, G.J. Koehler. A new direction in group support system theory: an evolutionary approach to group decision-making. 1999b, URL: http://citeseer.ist.psu.edu/rees99new.html.

G. Syswerda. Uniform crossover in genetic algorithms. In: *Proceedings of the Third International Conference on Genetic Algorithms*, 2–9, 1989.

A Min-Max Genetic Algorithm with Alternating Multiple Sorting for Solving Constrained Problems

Timo Mantere
Department of Electrical Engineering and Automation
University of Vaasa
FIN-65101 Vaasa
timan@uwasa.fi

Abstract

This paper introduces a min-max genetic algorithm that can naturally be applied to the min-max problems. A min-max genetic algorithm was originally designed for simultaneous minimization and maximization of the same object function during the same optimization run. In this paper we apply the method with multiple sorting for optimizing constrained functions. The other end of the genetic algorithm population minimizes the constraint violations and the other end maximizes the values of feasible solutions. According the results this method reaches the feasible area reasonably fast and consistently and produces relatively good results.

1 Introduction

This paper studies if the use of min-max genetic algorithm (MMGA) is beneficial when optimizing constrained functions. The MMGA means genetic algorithm where the elitism is applied to the both ends of the population fitness scale. In other words the number of individuals with the lowest fitness value and number of individuals with the highest fitness value will survive to the next generation. In practice this means that we are simultaneously minimizing and maximizing the same object function.

In an elitist steady-state genetic algorithm (GA) the optimization direction depends on the elitism. If some individuals with the highest fitness values will survive to the next generation we are maximizing the target function and vice versa if a number of individuals with lowest fitness values are we are minimizing the problem. If we preserve both individuals with the highest fitness value and individuals with the lowest fitness values we have MMGA, which does simultaneous minimization and maximization of the same object function. This multi-objective, or more precisely bi-objective, optimization method could also be called two-headed GA or reciprocal GA (Mantere, 2004).

This method was first introduced in papers (Mantere and Alander, 2001 and 2002). However, the application therein was relatively special, so the somewhat promising result with that application could not be generalized. In paper (Mantere, 2004) this method is represented more precisely with a set of unconstrained optimization problems. The paper reached the conclusion that this method is beneficial if the problem is difficult, multimodal and non-serialisable. The most important requirement seemed to be that two or more of the problem parameters had must joint effect and influence together to the component of the fitness value sum. With completely separable problems like onemax where each parameter alone directly produces one component of the total fitness value sum, the use of MMGA was not beneficial.

The findings in (Mantere, 2004) also seem to support the claim that MMGA is beneficial when one needs to find both the minimum and maximum of a problem simultaneously. There is a set of problems called min-max problems where this kind of simultaneous minimizations and maximizations are needed.

In this paper we apply the min-max GA method to a collection of constrained test functions in order see if this method works with them. We treat constrained maximization problem as a multi-objective problems and MMGA is applied so that the same genetic algorithm maximizes the object function and minimize the constraint violations. This is achieved by multiple sorting, where the feasible solutions are first sorted to the other end of fitness scale according to the fitness value. Thereafter the infeasible solutions are sorted to the other end, according to the constraint violations.

This way we can have simultaneous optimization of feasible solutions and the minimization of constraint violations among the infeasible solutions in the same GA population.

1.1 Genetic Algorithms

Genetic algorithms (Holland, 1992) are computer based optimization methods that uses the Darwinian evolution of nature as a model. The solution base of the problem is encoded as individuals that are chromosomes consisting of several genes. The GAs are simplified models derived from the natural genetics, *e.g.* in GAs the individual (phenotype) is usually exactly derived from the chromosome (genotype), whereas in the nature phenotype is not directly derived genotype, but the age, living conditions, diseases, accidents etc. of an individual have effect to the phenotype. In GAs the virtual individuals are tested against the problem represented as a fitness function. The better the fitness value individual gets the better chance it has to be selected to be a parent for new individuals. The worst individuals are removed from the population in order to make room for the new generation. Using crossover and mutation operations GA creates new individuals. In crossover we select the genes for a new chromosome from each parent using some reselected practice, *e.g.* one-point, two-point or uniform crossover. In mutation we change random genes of the chromosome either randomly or using some predefined strategy. The GA strategy is usually elitist and follows the "survival of the fittest" principles of Darwinian evolution.

1.2 Related work

The MMGA method is relatively simple, but we have not yet found other studies suggesting similar method. The book by Bäck *et al* (2000) introduces several ways of enhancing the performance of evolutionary algorithms, but do not represent a report of similar experiments as ours.

That might be due that there is a very limited set of problems where this kind of simultaneous minimization and maximization of the same function is necessary or profitable. Perhaps, someone has tried this kind of method earlier with the problems it is not well suited for, but results have not been good enough to be published (Alander, 2004).

This method has some similarity to co-evolution (Brodie, 1999, Bäck *et al*, 2000), but in this case we have only one species, with two races, that evolves simultaneously towards both "bad" and "good" solutions (= minimum and maximum) for the target problem.

In multi-elitist GA methods (Bellamo *et al*, 2002) elitism is applied so that in multi-modal problem the best individual from each peak is preserved to the next population. In our method we preserve peak and valley.

Obviously other kind of methods of multi-objective optimizations (Bäck *et al*, 2000, Deb *et al*, 2002) can be seen as related work to this one, since we also have two objectives here.

1.3 The benchmark problems

The set of constrained benchmark functions used in this paper have earlier been used by many researchers, we compare our results with the three papers (Runarsson and Yao, 2000, Mezure-Montes and coello, 2005, Li *et al*, 2005) that represented the best results with evolutionary algorithms we found.

These test problems are introduced in the paper by Runarrson and Yao (2000), where each function formula are given and the characteristics of the corresponding problem are described.

We found several other papers, suggesting several other methods for optimizing these functions, but many of these papers represented their results with only 4 to 7 problems of the 13 benchmark problems. We feel that leads to the biased and weak conclusions. We reach the optimum with 5 of the 13 benchmark problems. If we would have chosen to report our results only with those five, we would have misrepresented the potentials of our method. We think it is necessary to also represent how it works with the other 8 benchmark problems.

There is a paper by Morovic and Wang (2003) that clearly demonstrates the importance of not using a limited benchmark set when trying to represent the results with some algorithm. They showed that by selecting convenient five-image subset from the set of 15 test images, they could get any of the six Gamut Mapping Algorithms they tested to appear like it is the best when compared to the others. They looked 21 research papers on GMAs and find out that all these papers have represented results by using only 1 to 7 test images from the 15 image test set.

1.4 Some discussion about the method

In order to have co-evolution the species should have some interaction. In this case one might suspect that there is no beneficial interaction, since the result of crossover between "bad" and "good" population members are unlikely to result any good solutions neither to the maximization or minimization of the problem. However, in this case we do have two races of the same species in the same population. They are of the same species, since they can mate with each others, and their genetic structure is the same. They are of different race, since they have one profound difference; the other end of the population is usually occupied by feasible solutions (race1), and another with unfeasible solutions (race 2).

Flores and Smith (2003) conclude that the presence of infeasible solutions speeds up the search. So, the interaction between feasible and infeasible solutions should be beneficial.

The main reason that the presence of infeasible solutions in the population is beneficial is that most of the test problems are such that the optimum is located in the border of infeasible area.

When we minimize the sum of constraint violations we are moving towards the feasible area. However, we should reach the feasible area from the different directions, since both the search for the feasible area and the search of optimum can get stuck to the local optimum. Therefore using different or alternating strategies and target functions becomes reasonable.

2 The proposed method

In the constrained optimization heuristic method such as a genetic algorithm is likely to generate a lot of infeasible solutions that violate the constraints. The usual way to handle constraints is to add penalty term to the fitness function, so that it penalizes the constraint violation. It is difficult to define a proper penalty function and thus one often either over penalize or under penalize. In over penalized situation all feasible solutions are considered better than infeasible. This usually means that the possibly beneficial genetic information that infeasible solutions may possess will quickly disappear from the population and we will get stuck to the local optima. In the case of under penalizing the penalty is not hard enough and all our solutions may violate constraints.

In this paper we do not want to use any penalty functions. Instead we use multiple sorting rules in order to sort the population, so that the feasible and infeasible solutions will remain in the same population. One may argue that using the sum of constraint violations is the same thing as using the penalty function. We think that it should rather be seen as using the minimization of constraint values also as objectives.

When we do changes to the sorting rules it means that different infeasible solutions become the strong individuals among the population. It also means we are moving towards the feasible area from the different direction.

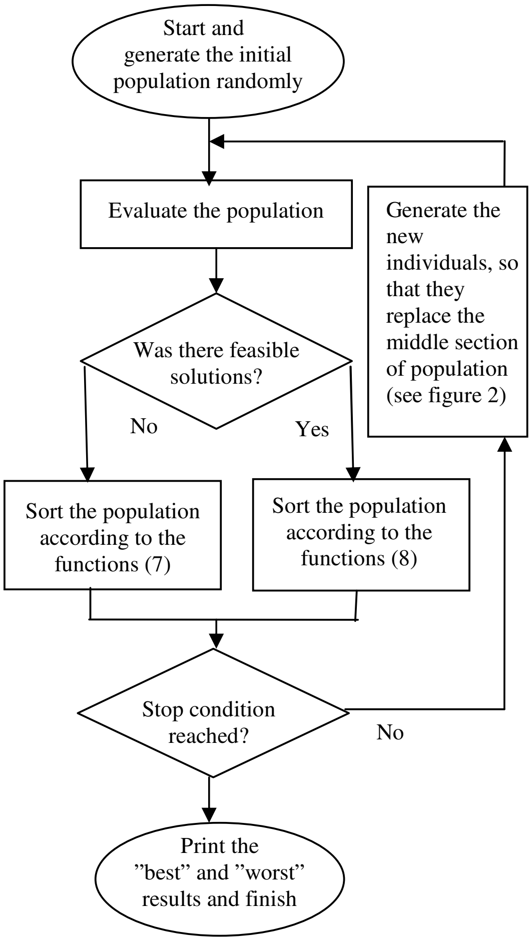

Start and generate the initial population randomly
Evaluate the population
Was there feasible solutions?
No
Yes
Sort the population according to the functions (7)
Sort the population according to the functions (8)
Stop condition reached?
No
Generate the new individuals, so that they replace the middle section of population (see figure 2)
Print the "best" and "worst" results and finish

Figure 1. The flow chart of the proposed method, MMGA with multiple sorting.

We have the constrained problem:

$$Maximize\ f(x), \quad x = (x_1,...,x_n) \in R^n \qquad (1)$$

with box coefficients that limit the search space:

$$x_{i_{\min}} \le x_i \le x_{i_{\max}}, \quad i = \{1,...,n\} \qquad (2)$$

and with the inequality coefficients:

$$g_j(x) \le 0, \quad j = \{1,...,m\} \qquad (3)$$

The equality constraints:

$$h_k(x) = 0, \quad k = \{1,...,q\} \tag{4}$$

are transformed into an inequality constraints by defining that:

$$g_j(x) = |h_k(x)| - \delta \leq 0 \tag{5}$$

That way all the constraint violations will be positive. If the target function is minimization problem, we chance the sign and get maximization, so we are minimizing the constraints violations and maximizing the function value.

In the proposed method we use steady-state genetic algorithm (fig. 1) with some amount of elitism N_e, in such a way that $N_e/2$ of the best individuals and $N_e/2$ of the worst individuals will survive to the next generation (figure 2). In practice this means that the population should evolve towards the both directions, the global maxima and minima.

The parent selection scheme is such that each of the individuals in the population has equal opportunity to become parents, $1/N$. This was chosen because if we use some kind of emphasis on good individuals, like ranking order or fitness value, as a base for unequal mating opportunity, only the "bad" and "good" individuals would mate, and the middle ones would get a very little chance to mate. We also compare every new individual against the members of old population and the other new individuals, so that no identical individuals exist in the population.

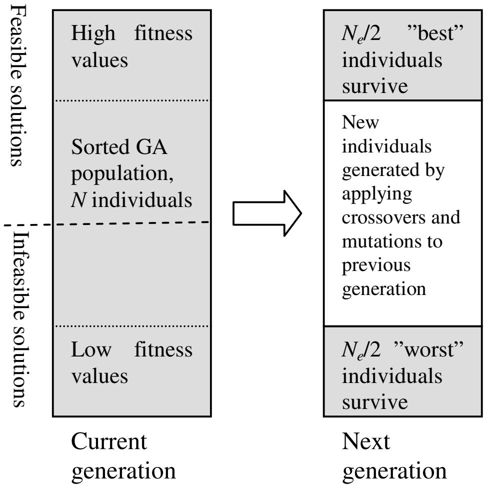

Feasible solutions
Infeasible solutions
High fitness values
Sorted GA population, N individuals
Low fitness values
Current generation
Ne/2 "best" individuals survive
New individuals generated by applying crossovers and mutations to previous generation
Ne/2 "worst" individuals survive
Next generation

Figure 2. The elitism scene of the proposed method, both a number of individuals with highest fitness values and lowest fitness values will survive to the next generation.

We use multiple sorting with two different settings. In the beginning of the optimization, before our algorithm reaches the feasible area (one or more individuals of the population are feasible solutions for that problem), we sort the whole population according to the first function of the function pair (7), and after that the bottom half of the population [$N/2$, N] by the second function of pair (7). The function pair is selected randomly in every generation; so that each function pair has 25% chance to be the pair that biases the optimization direction.

When the feasible area is reached, all the feasible solutions are first sorted according to the function value $f(x)$, the rest of the population or at least the bottom 25% [$3N/4$, N] is sorted according to one of the four alternating functions seen in (8). The other function that minimizes the other end is again randomly chosen in every generation, and each of the four functions has the 25% chance to be selected. Two of these functions use only the coefficient violations for biasing our search, and the other two use different combinations of function value and coefficient violations.

We decided to use target function that returns us three values:

$$\begin{cases} f(x) \\ \sum_{j=1}^{m} g_j(x) \\ Max\ \{g_j(x)\} \end{cases} \tag{6}$$

They represent the target function value, the sum of all constraints violations, and the highest constraint violation accordingly.

These values are used in different functions pairs that we decided to use for sorting the population before reaching the feasible area, they are:

$$\begin{cases} Maximize\colon f(x), \quad Minimize\colon \sum g_j(x) \\ Maximize\colon -Max\{g_j(x)\}, \quad Minimize\colon \sum g_j(x) \\ Maximize\colon f(x) - \sum g_j(x), \quad Minimize\colon Max\{g_j(x)\} \\ Maximize\colon f(x) - \sum g_j(x), \quad Minimize\colon \sum g_j(x) \end{cases} \tag{7}$$

Different sorting function biases the search direction differently. Each function pair has at least one function that leads the search towards feasible area. The other function may be just lead towards the higher function $f(x)$ values, but it may also lead toward the feasible area from the different direction. The reason for using the multiple sorting is to force the population ends differ from each other; so that the other end contains the infeasible solutions and hopefully the other end will contain feasible ones.

Table 1. The amount of parameters, *n*, and the amount of inequality constraints, *m*, of each test problem g01 to g13. Also the number of generations (Best, Median, and Worst) that MMGA needed to reach the feasible area of each function, and also the descriptive statistics of 30 MMGA test runs with each problem. * Marks the average of 21 out of the 30 test runs that reached feasible area.

Test problem			Reach feasible			Optimal solution	Descriptive statistics of the MMGA results				
g	*n*	*m*	B	M	W		Best	Median	Average	Worst	St.Dev.
01	13	9	42	61	329	-15.000	-15.000	-15.000	-15.0000	-15.0000	0
02	20	2	1	1	1	0.803619	0.80355	0.793961	0.79185	0.76144	0.01100
03	10	1	13	33	87	1.000	1.0000	0.99997	0.99980	0.99823	0.00044
04	5	6	1	1	1	-30665.539	-30665.487	-30665.243	-30665.043	-30664.060	0.43048
05	4	5	423	3375	Infea	5126.498	5126.503	5290,247	5378.786 *	Infeasible	275.349
06	2	2	5	21	50	-6961.814	-6961.814	-6961.814	-6961.814	-6961.814	0
07	10	8	19	33	60	24.306	24.448	25.986	25.589	29.509	1.237
08	2	2	1	1	6	-0.095825	-0.0958	-0.0958	-0.0958	-0.0958	0
09	7	4	1	2	6	680.630	680.643	680.684	680.691	680.791	0.03422
10	8	6	25	97	460	7049.331	7054.002	7607.787	8034.110	10346.330	1477.12
11	2	1	14	62	194	0.750	0.750	0.750	0.750	0.750	0
12	3	729	1	1	1	-1.000	-1.000	-1.000	-1.000	-1.000	0
13	5	3	66	146	56	0.05395	0.06629	0.86610	0.77731	0.99041	0.23066

When the feasible area is reached our function pairs changes to:

$$\left\{ \begin{array}{ll} & Minimize : \sum g_j(x) \\ Maximize : f(x), & Minimize : Max\{g_j(x)\} \\ & Minimize : \sum g_j(x) - f(x) \\ & Minimize : Max\{g_j(x)\} - f(x) \end{array} \right. \qquad (8)$$

The reason for changing the sorting functions randomly from generation to generation is to make more unlikely that our optimization would get stuck to the local minimum. When the sorting function changes among the infeasible solutions, the different members of the population becomes more fit, and that biases the search direction differently, so we are entering the feasible area from the different direction. This procedure also seems to reduce the time needed before the algorithm reaches the feasible area.

3 Experimental results

In order to test the proposed method we decided to use a real-value coded GA with uniform and arithmetic crossovers and Gaussian mutation. The size of the GA chromosome depends on target function, *i.e.* how many parameters (*n*) it has (table 1).

First, we run some preliminary tests in order to find a set of good GA parameters for these problems. We did not found any universal parameter set that would work with all of the problems, so we selected the following: population size N=100, elitism N_e=50, mutation percentage 50/*n* (table 1) with Gaussian mutation and crossover ratio 50 (both uniform and arithmetic crossovers with 50/50 chance), execute 6999 generations. This means total 350 000 function evaluations (100+6998*50). The high elitism was used because it forces the two races in the population interact, having interracial crossovers, more efficiently. High mutation percentage, since the amount of average Gaussian mutation was small.

Our test function set in these experiments was taken from study of Runarsson and Yao (2000). The limits of optimization space (box coefficients for X_i) were also adopted from them, as well as the degree of violation δ=0.0001 for the equality constraints $|h(x)|- \delta<=0$. They report that they used 200*1750 (=350 000) function evaluations, and they used 30 test runs with each problem, and so did we. Our results in this paper are compared to their results with all of the 13 problems, and also against the results represented by Li *et al* (2005) and by Mezures-montes and Coello (2005).

Table 1 shows how quickly our MMGA with multiple alternating sorting methods reached the feasible area with each of these problems (Best, Median and Worst case of the 30 test runs). With the problem g05 we reached the feasible area only 21 times out of 30 test runs, so the worst result was infeasible. With the all other problems our method reached the feasible area between 1 to 460 generations (that corresponds from 1 to 23000 trials).

Table 2. The best, average and worst results with our method (MMGA), and with methods represented by Runarsson and Yao (RY, 2000) Li *et al* (LJW, 2005), and Mezura-Montes and Coello (MC, 2005) with different test functions.

	Best results				Average result				Worst result			
g	MMGA	RY	MC	LJW	MMGA	RY	MC	LJW	MMGA	RY	MC	LJW
01	-15.000	-15.000	-15.000	-15.000	-15.000	-15.000	-15.000	-15.000	-15.000	-15.000	-15.000	-15.000
02	0.80355	0.80352	0.80360	0.80360	0.79185	0.78198	0.78523	0.78571	0.76144	0.72629	0.751322	0.74448
03	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.998	1.000	1.000	1.000
04	-30665.49	-30665.54	-30665.54	-30665.54	-30665.04	-30665.54	-30665.54	-30665.54	-30664.06	-30665.54	-30665.54	-30665.54
05	5126.503	5126.497	5126.599	5126.498	5378.786	5128.881	5174.492	5133.910	Infeas.	5142.472	5304.168	5165.659
06	-6961.814	-6961.814	-6961.814	-6961.814	-6961.814	-6875.940	-6961.284	-6961.810	-6961.814	-6350.262	-6952.482	-6961.780
07	24.448	24.307	24.327	24.312	25.589	24.374	24.475	24.498	29.509	24.642	24.843	24.825
08	-0.0958	-0.0958	-0.0958	-0.0958	-0.0958	-0.0958	-0.0958	-0.0958	-0.0958	-0.0958	-0.0958	-0.0958
09	680.643	680.630	680.632	680.630	680.691	680.656	680.643	680.650	680.791	680.763	680.719	680.696
10	7054.00	7054.316	7051.903	7049.499	8034.110	7559.192	7253.047	7168.782	10346.33	8835.655	7638.366	7474.948
11	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750
12	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000
13	0.06629	0.053957	0.053986	0.053986	0.77731	0.05701	0.166385	0.055683	0.99041	0.21692	0.468294	0.06171

The table also contains the descriptive statistics of our test runs. They show that our method reached the optimal solution in every test run with the problems g01, g06, g08, g11, and g12.

Table 2 shows the comparison of our results: best, average, and worst values to the results represented by Runarsson and Yao (RY, 2000) with their evolutionary strategy with stochastic ranking, and by Li *et al* (LJW, 2005) with GA hybrid with interpolation, and by Mezura-Montes and Coello (MC, 2005) with multimembered evolution strategy.

The comparison shows that all of the compared methods reached the optimum every time with the problems g01, g08, g11 and g12. With the problem g06 our method was the only one that reached the optimal solution every time. With the problem g04 all the other methods except ours reach the optimum every time.

Our method works well with the problem g02, where our average and worst results were better than those got by others. With the problem g03 our worst result was worse than those got by others, but the best and average was as good. With g09 our method did not reach optimum and performed slightly worse than other methods.

The most difficult problems for our method seem to be the test problems g05, g07, g10 and g13. With these problems the best result with our method was close to the results got by others, but the average and worst results of 30 test runs were much worse.

With g05 our method also failed to reach feasible area 9 times out of 30.

4 Conclusions and future

In this paper we studied if MMGA (min-max genetic algorithm) is worth of considering in special problems where we need to find both the function minima and maxima. In the case of constrained problem we can define the function so that the same GA maximizes values of the feasible solutions and the other end minimizes the constraint violations of infeasible solutions. In this paper that was achieved by multiple sorting, where sorting rules alternate randomly.

We used conditional fitness function that changes when feasible area was reached, *e.g.* in the beginning, the both ends of population are used for speed up the search of feasible area. When feasible area is reached the other end will start to maximize the feasible solutions, and the other end is dedicated for minimizing either the sum or maximum constraint violations or their sum with the target function value.

The results show that the MMGA method with multiple sorting works relatively well with these constrained test problems. It reaches the feasible region relatively fast and consistently. Unfortunately with one of the benchmark problems we had some difficulties to reach feasible area. We need to study the characteristics of that problem more precisely in order to find out the reason.

Our results were relatively good when compared with the other methods, but our method still needs some fine tuning, since with four of the benchmark problems, our average and worst results were not good. We need to do further analyze of what characteristics of these problems causes the obstacles for our method, and improve our method accordingly.

Acknowledgements

The research work of the author of this paper was funded by the research grant from the TietoEnator fund of the Finnish Cultural Foundation.

References

Alander, J.T. *Personal comments*. 10.6. 2004.

Bäck, T., Fogel, D.B., and Michalewitz, T. (eds.). *Evolutionary Computation 2 Advanced Algorithms and operators*. Institute of Physics Publishing, Bristol and Philadelphia, 2000.

Bellomo, D. Naso, D., and Turchiano, B. Improving genetic algorithms: an approach based on multi-elitism and Lamarckian mutation. In *IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, Hammanet, Tunisia, 6-9, Oct. 2002: 89-94, 2002.

Brodie E. III, and Brodie E. Jr. Predator-prey arms races. *BioScience* **49**: 557-568, 1999.

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* **6**(2): 182-197, 2002.

Duarte Flores, S., Smith, J. Study of Fitness Landscapes for the HP model of Protein Structure Prediction. In *The 2003 Congress on Evolutionary Computation CEC '03* Volume 4, 8-12 Dec., IEEE Service Center, Piscataway, NJ: 2338–2345, 2003.

Holland, J. *Adaptation in Natural and Artificial Systems*. The MIT Press, Cambridge, MA, 1992.

Li, H., Jiao, Y.-C., and Wang, Y.: Integrating the simplified interpolation into the genetic algorithm for constrained optimization problems. In Hao et al. (eds) *CIS 2005, Part I, Lecture Notes on Artificial Intelligence 3801*, Springer-Verlag, Berlin Heidelberg: 247-254, 2005.

Mantere T. Reciprocal elitism in GA optimization. In J. Alander, P. Ala-Siuru, and H. Hyötyniemi (eds.), *Step 2004 – The 11th Finnish Artificial Intelligence Conference, Vol. 3, Origin of Life and Genetic Algorithms*, Heureka, Vantaa, 1-3 September 2004, Finnish Artificial Intelligence Society, Helsinki: 151-160, 2004.

Mantere, T., Alander, J.T. Developing and testing structural light vision software by co-evolutionary genetic algorithm. In *QSSE 2002 The Proceedings of the Second ASERC Workshop on Quantative and Soft Computing based Software Engineering*, Feb 18-20 2002, Banff, Alberta, Canada. Alberta Software Engineering Research Consortium (ASERC) and the Department of Electrical and Computer Engineering, University of Alberta: 31-37, 2002.

Mantere, T., Alander, J.T. Testing a structural light vision software by genetic algorithms – estimating the worst case behavior of volume measurement. In D. P. Casasent and E. L. Hall, eds., *Intelligent Systems and Advanced Manufacturing: Intelligent Robots and Computer Vision XX: Algorithms, Techniques, and Active Vision*, volume SPIE-4572, Newton, MA, October 29-31 2001, SPIE, Bellingham, Washington, USA: 466-475, 2001.

Mezure-montes, E., Coello, C.A.C. A simple multimembered evolution strategy to solve constrained optimization problems. *IEEE Transactions on Evolutionary computation* **9**: 1-17, 2005.

Morovic, J., Wang, Y. Influence of test image choice on experimental results. In *Proceedings of 11th Color Imaging Conference*, Scottsdale, AR, Nov. 3-7: 143-148, 2003.

Runarsson, T.P., and Yao, X.. Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation* **4**(3): 284-294, 2000.

On fitness distance distributions and correlations, GA performance, and population size of fitness functions with translated optima

Janne Koljonen
Department of Electrical Engineering and Automation, University of Vaasa
P.O. Box 700, FIN-65101, Vaasa, Finland
jako@uwasa.fi

Abstract

Fitness distance correlation has been proposed as an analysis tool for the fitness function landscape of genetic algorithms. It has been shown that it works in some cases, but there are also counter-examples. We studied the fitness distance correlation and the optimal population size by two fitness functions with a translational minimum and with a meta-genetic algorithm. We discovered that the fitness distance distribution and the actual optimization performance depend strongly on the place of the optimum. We also found that fitness distance correlation explains moderately the variations in the performance of the genetic algorithm and also some of the variations in the optimal population size obtained by a meta-genetic algorithm.

Keywords: elitism, fitness distance correlation, meta-GA, optimal GA, population size.

1 Introduction

A major effort has been given to solve when evolutionary computing, i.e. evolutionary strategies (ES), evolutionary programming (EP) and genetic algorithm (GA), will be effective at finding the global optimum. The problem is challenging due to the nonlinear and stochastic nature of the algorithms, and because evolutionary algorithms is actually a family of algorithms, from which the applier selects a proper one with appropriate parameters and operators according to the application.

There are two separate classes to approach the performance analysis: one that tries to solve the efficiency without actually running the algorithm and without knowing the solution, and another one that analyzes the performance after first obtaining the solution. The first approach would be very useful if it could be done with less effort than the runs of the algorithm would require, but it is probably harder to achieve an effective method to that than to the second case.

The methods to analyze and predict ES, EP and GA performance (and the hardness of the objective function determining, together with the operators etc, the performance of the algorithm) developed so far are based on the building block theory (Forrest and Mitchell, 1993; Goldberg, 1989), multimodality (Rana, 1998), fitness distributions (Borenstein and Poli, 2004; Popovici and De Jong, 2003), fitness distance correlation (Jones and Forrest, 1995; Vanneschi and Tomassini, 2002), and several others, too. Each of them can explain some phenomena but, unfortunately, not all. There won't probably be any single method that would explain every case and hence this many are needed.

The main problem of some of them is that they don't take into account the genotype-phenotype mapping of the problem and the genetic operators themselves that actually define the dynamics of the evolutionary algorithm. Modifying or changing them can change the problem completely. The analysis tools should therefore be based on the operators themselves (Altenberg, 1997). There are also analyses methods that take some of the dynamic properties into account (See e.g. Syswerda, 1989). With fixed operators a static analysis of the fitness distance correlation can though be adequate in some cases, as will be shown in this paper.

2 Fitness distance distributions

Genetic algorithms work with the aid of the information given by the fitness function. The usefulness of this information depends on the genotype-phenotype mapping, fitness landscape, and the genetic operators. The fitness function works on phenotype while the GA itself on genotype. This causes that the fitness function does not directly tell, how close to a solution the GA is. It can even be misleading the algorithm away from the solution, in terms of the Hamming distance of the binary genotype representation to the optimum.

If the global optimum (or optima) is known, the Hamming distance to the optimum (or to the nearest global optimum) and fitness value of a genetic individual can be computed. If possible and sensible, in terms of computational efforts, we can calculate distances and fitness values of all possible genotypes to obtain a fitness distance distribution (FDD); otherwise a random sample should be obtained. Jones and Forrest (1995) have plotted Hamming distances against fitness values as scatter plots. In our opinion, 3-dimensional frequency plots are in many cases more illustrative.

Generally, with some exceptions (Altenberg, 1997), the regular GA is effective if the Hamming distance becomes smaller as the fitness becomes better, be it higher or lower whether we are maximizing or minimizing. In this paper, we are always minimizing and at the only optima of the test problems the fitness function attains 0.

The exceptions in (Altenberg, 1997) are obtained using single-point crossover. Uniform crossover (Syswerda, 1989) is probably the easiest crossover operation to handle theoretically and in many cases the most efficient one. With uniform crossover it is easily seen that crossing two individuals with low Hamming distances with respect to the optimum has a better chance to produce an offspring representing a solution than individuals of higher Hamming distances. Individuals with low distance may not necessarily have low fitness values, as can be seen e.g. in Figure 3. On the other hand, there may be genotypes with low distance and high fitness that are blocked from the elite as well as genotypes with high distance and low fitness i.e. *false friends*. As the GA tries to minimize the fitness in the latter case, it may on the contrary create more individuals false friends thus lowering the chance to obtain a solution.

The properties of the fitness-distance dependence and distribution can be examined with 3D frequency plots or scatter plots. Jones and Forrest have suggested (1995) that this information could be summarized into the Pearson correlation coefficient *r*, called fitness distance correlation (FDC):

$$r = \frac{c_{FD}}{s_F s_D}, \tag{1}$$

where c_{FD} is the covariance of the samples of the fitness and distance pairs, and s_F and s_D are the standard deviations of the fitness and distance, respectively.

The fitness distance correlation has been used at least for genetic algorithms (Jones and Forrest, 1995; Altenberg, 1997) and genetic programming (Vanneschi *et al*, 2002; Vanneschi and Tomassini, 2003). It has been shown by some counter-examples that FDC is not always a good estimate of problem difficulty. FDC has been mainly tested on the common benchmark problems and with other not particularly interrelated problems. In this paper, we study families of closely interrelated problems with FDC and compare FDC to the GA performance.

3 Genetic algorithm

Genetic algorithms is a stochastic optimization method effective for difficult and nonlinear problems (Alander, 1994; Alander, 2002). We used a GA with two genetic operators: uniform crossover (Syswerda, 1989) and bitwise mutation. The GA had two populations: one with mature individuals, also known as elite, from which the parents were randomly drawn, and a population of immature individuals into which the offspring are temporarily placed after creation. To refer to the size of the populations we use symbols n_e and n_i, respectively. Hence, the total population size is $n_e + n_i$. By relative elitism we mean $n_e/(\ n_e + n_i)$. n_{max} denotes for the maximum number of trials, inclusive the original population, allowed for the GA to do.

Each individual in the elite had an equal probability to be selected as a parent, but the parents in the crossover had to be separate to avoid clones. We used natural binary coding in genotype-phenotype mapping.

4 Test problems

To study the FDC as a GA performance predictor we formed families of problems that have a transla-

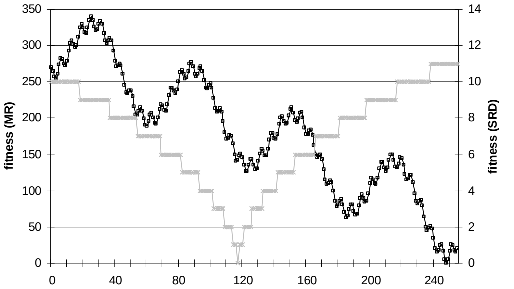


Figure 1: Fitness landscape of the 8-bit MD_{248} (black rectangles) and SRD_{117} (gray crosses).

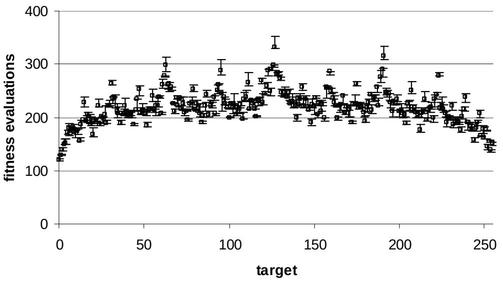


Figure 2: Minima, medians and maxima of numbers of evaluations of three runs that averaged 1000 GA runs (n_{max} = 800) on each target of SRD. The population sizes were fixed (n_e = 32, n_i = 16).

tional optimum. The first fitness function (FF) was the square-root of the distance (SRD) to the target rounded down to the nearest integer. By changing the target the problems (denoted by subscripts, e.g. SRD$_0$) the fitness landscape changes. The fitness function is thus:

$$fitness = \left\lfloor \sqrt{|phenotype - target|} \right\rfloor . \qquad (2)$$

To validate the results another test function (modified Rastrigin, MR) with a multimodal phenotype landscape and a translated optimum was used:

$$fitness = \lfloor |x\text{-}t| \text{-}10\cos(x\text{-}t) \text{-}50\cos((x\text{-}t)/10) + 60 \rfloor . (3)$$

where x is the phenotype presentation of the bit vector and t is the target, where *fitness* = 0.

4.1 Fitness landscape

The fitness landscapes (phenotype) of the test functions are shown in Figure 1, where 8 bits are used.

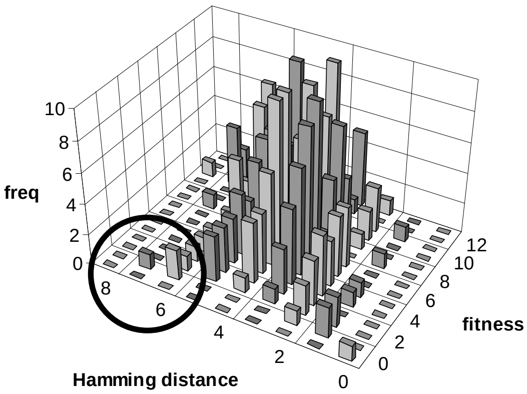


Figure 4: Fitness distance distribution of one of the hardest problems of FF the family (SRD$_{63}$). There are samples whose fitness is 1 but the Hamming distance large (in the circle).

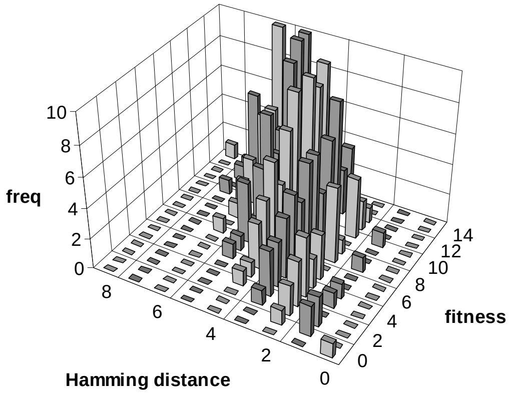


Figure 3: Fitness distance distribution of the SRD$_0$. This problem was the GA easiest of the fitness function family.

With a phenotype search strategy the SRD problems seems to be quite easy, but the MRs do not.

4.2 Fitness distance distributions

We started to study the fitness functions running the GA with all possible target values; this was considered reasonable with 8-bit fitness functions. The results for SRD are shown in Figure 2. The figures for MRs (not shown) look essentially the same.

With these preliminary runs we found that certain targets differ significantly from others. The easiest ones were near the limits of the range; 0 was the easiest. The most difficult ones were the following targets: 63, 95, 126, 127, and 191. Each of them has false friends. E.g. if the solution is 127_{10} = 01111111_2, then 128_{10} = 10000000_2.is clearly a false friend, whose Hamming distance to 127 is 8, but whose fitness is low with continues fitness functions.

Let us examine the fitness distance distributions of the fitness function with some target values. We

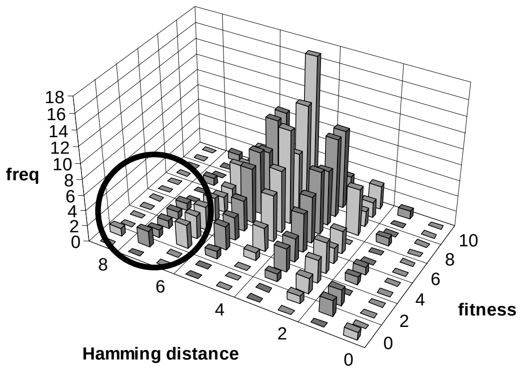


Figure 5: The hardest problem of the ff family (SRD$_{127}$) has a flatter distribution and many false friends (circled) predicting poor evolvability.

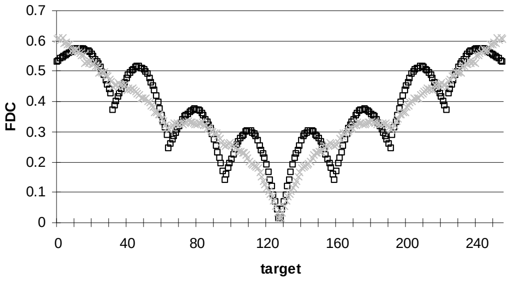


Figure 6: Fitness distance correlation with different targets. Black rectangles = MR, gray crosses = SRD.

can then compare the distributions of the GA hard ones to the easier ones.

The distribution in Figure 3 belongs to the easiest one, as for our GA. It has a clear positive correlation between the Hamming distance and fitness and several low distance low fitness genotypes, from which the solution is easily obtained by crossover.

The FDD in Figure 4 is in turn one of the GA hardest ones. The correlation seems quite strong, may it be less that in Figure 3, but probably the several false friends not existing in Figure 3 cause the GA sometimes to be stuck.

The hardest fitness function of the family was when target was 127. Its fitness distance distribution shows low correlation as well as many false friends. Yet SRD_{63} (Figure 4) was nearly as difficult as SRD_{127} even though its FDD looks much easier compared to the very flat and uncorrelated one in Figure 5.

4.3 Fitness distance correlation

W calculated all FDCs of the fitness function families (Figure 6) to find out how well they predict the GA performance. The results in Figure 6 confirm the conclusions made on the basis of fitness distance distributions, as for variation in FDCs.

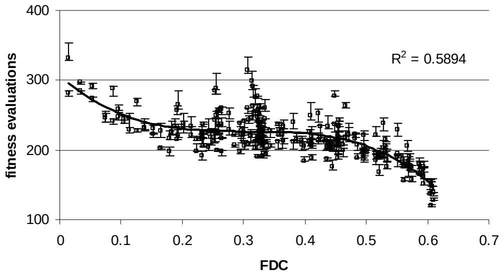


Figure 7: Correlation between the GA hardness and the FDC is moderate: with a 3rd degree polynomial the R^2 = 59 %. The population sizes and other parameters were fixed (n_e = 32, n_i = 16). Minima, medians and maxima of means of three repetitions of 1000 GA runs using SRD functions.

In Figure 7 the GA performances of SRD functions (as in Figure 2) are plotted against the FDCs. The plot shows only moderate correlation (R^2 = 59%) and some outliers, so the FDC seems to be inadequate predictor. For MR functions R^2 = 40 %. The reason (as will be shown) is that the population sizes (and other parameters, too) are kept fixed when changing the target.

5 Optimization speed and population size

It was presumed that different target values of the fitness function families require different population sizes as the FDC and GA hardness seemed to vary significantly. We tested this hypothesis by changing the population size parameters and running the GA 200 times. This time we used the 10-bit SRD function. The maximum number of new individuals was limited to 800.

5.1 Results and discussion

Running the GA with the easiest and the hardest problem of the SRD family gives a clear indication

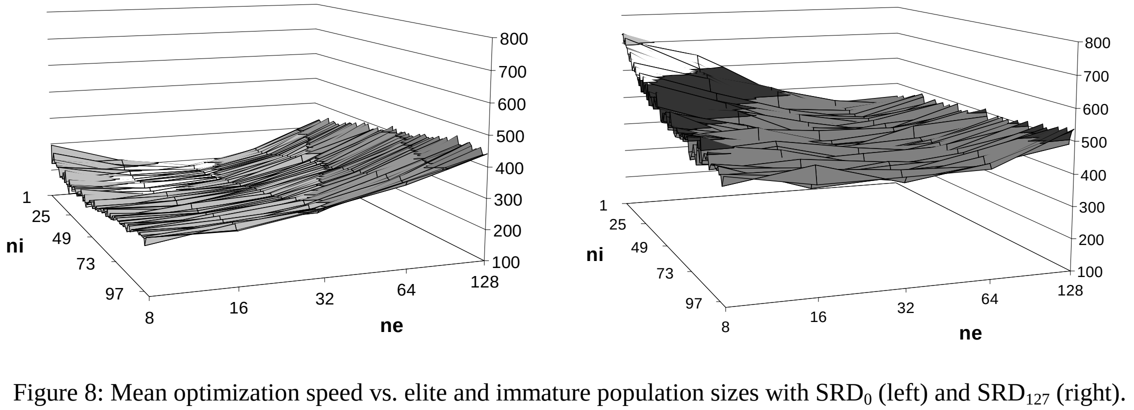


Figure 8: Mean optimization speed vs. elite and immature population sizes with SRD_0 (left) and SRD_{127} (right).

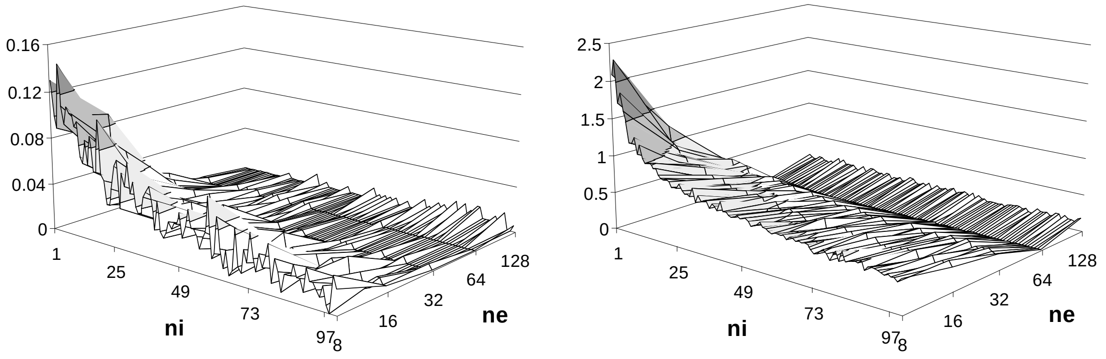


Figure 9: Mean of the best fitness of 200 runs vs. population sizes with SRD$_0$ (left) and SRD$_{127}$ (right).

how the population size must be tuned individually for each problem to get the optimal GA performance. Figure 8 depicts the mean GA performance vs. population sizes for the easiest and hardest SRDs. By comparison it is evident that the optimal population size increases when having harder problems. Moreover, if the population sizes were kept constant, depending which values were selected, the performance difference between targets varies.

6 Optimization reliability

The optimization speed is not everything but the reliability of finding a solution is also important. We measured the reliability by recording the best fitness of each run.

6.1 Results and discussion

We ran the easiest and hardest SRDs (Figure 9.) and results look very similar: the larger the population the better the reliability. However, the mean of the best fitness is over ten times worse for SRD$_{127}$ than for SRD$_0$, particularly with small populations.

7 Meta GA optimization

It seemed that fitness distance correlation was not very efficient in GA performance prediction. Our hypothesis was that this was due to non-optimal GA parameters, particularly population size, whose optima were shown to differ a lot from target to target. To continue studying FDC we optimized the population size parameters with a meta-GA.

7.1 Method

The optimization of GA parameters can be accomplished by another GA. These parameters include crossover type, mutation rate, population size etc. In other words, the parameter selection problem is formulated as another optimization problem (Mansfield, 1990; Alander, 2002; Bäck, 2002). It has been reported that the GA can optimize its own parameters, too (Hartono, 2004).

In meta-GA the individuals are GAs that solve the actual problem (See Figure 10). The fitness function is defined by measuring the GA performance: it could be for instance the number of fitness function calls required to find a solution. Each fitness function evaluation of a given meta-GA trial may give different results due to the stochastic nature of GAs. Thus the mutual fitness order of the meta-GA individuals in elite may change during optimization.

7.2 Results and discussion

We used a meta-GA to optimize n_e and n_i, whose ranges were determined so that the optimal values obtained from Figure 8 were included. The meta-GA parameters were fixed: n_{max} = 100, n_e = 32, and n_i = 4. The fitness function of the meta-GA was the number fitness function calls averaged over 100 GA runs using the same parameters. The actual GAs used the 8-bit SRDs or MRs and had n_{max} = 800.

The meta-GA was run three times with every 256 possible fitness functions. The fitness values of the

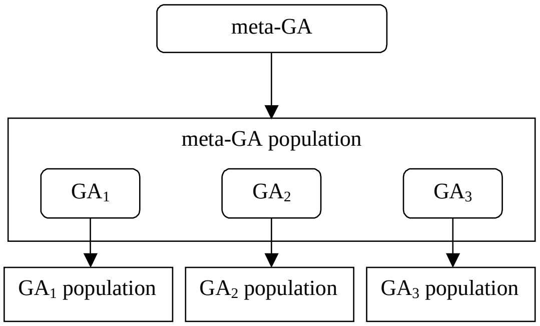


Figure 10: Principle of a meta-GA. Meta-GA optimizes the parameters of the actual GAs.

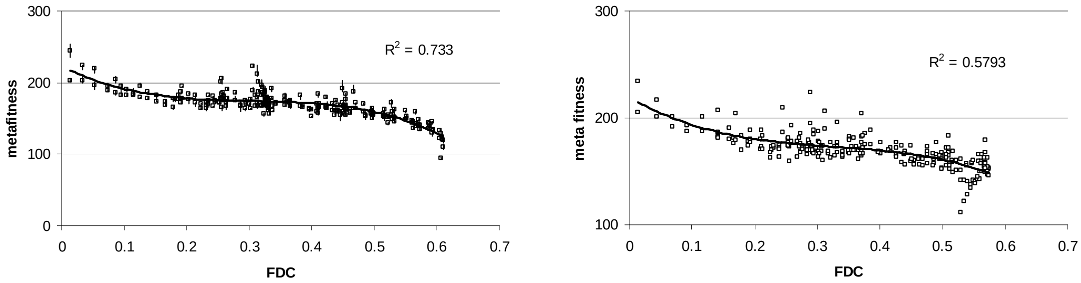


Figure 11: The mean best fitness of the last ten generations of the meta-GA chromosome vs. FDC for SRD (left) and MR (right). The 3rd degree polynomials explain 73 % and 58 % of the variation for SRD and MR, respectively (Compare the SRD result to 59 % with fixed population size in Figure 7). Min, med and max of three runs shown for SRD.

best meta-GA individuals are plotted against the corresponding FDC in Figure 11 (left for SRD, right for MR functions). The deviations of the three runs are also shown. The R^2 has risen from 59 % to 70 % by optimizing the population size.

Because the areas of optimal population sizes are relatively flat (See Figure 8) and the meta-GA fitness function is stochastic, there is a lot of internal variation in the population size. To overcome this we decided to use the average of the best chromosomes of the last ten generations of the meta-GA. By taking averages of the fitness value, elite size and immature population size, the stochastic variation was decreased.

Now the mean of the best fitness of the last ten meta-GA generations correlates a bit stronger with the FDC than when using only the best individual of the last meta-GA generation (Figure 11). The R^2 of the SRD and MR functions have risen from 59 % to 73 % and from 40 % to a reasonable 58 %, respectively.

A bigger improvement is obtained in the correlation between the optimal total population size (Figure 12) and the FDC for SRD function. However, the deviation of the optimal population size vs. FDC indicates statistical insignificance. Also total population size and meta-GA fitness correlate showing the well-known fact (Alander, 2002) that the smaller the population size, the faster the GA, unless the population is too small. The R^2 for population size vs. FDC are 43 % for SRD and 14 % for MR. The R^2 for population size vs. metafitness are in turn 65 % and 53 %, respectively.

8 Conclusions

One can usually predict the evolvability of a genetic algorithm by studying fitness distance distribution, despite some exceptions. The information can be compressed into fitness distance correlation (FDC). However, the method does not take into account the dynamics of the GA and there are counter-examples that show that low correlation does not necessarily imply poor GA performance.

We studied the ability to predict GA performance by FDC with two test functions. By translating the global minimum of the test functions whole families of similar functions were created. It was

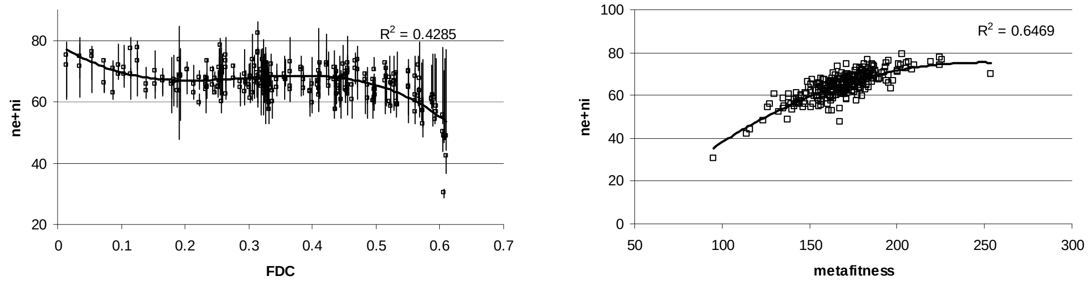


Figure 12: Correlations for SRD: Mean population size of the last ten meta-GA generations vs. FDC (left). R^2 for median runs is 43 %. Also the mean best fitness of the last ten meta-GA generations (right, deviations not shown) is correlated with the population size (R^2 = 65 %).

discovered that translation of minimum significantly changes fitness distance correlation and GA hardness. Fitness distance correlation was found a moderate predictor of the GA optimization speed when population size was fixed, but when optimizing the population size with a meta-GA and taking average of the best meta-GA individuals of the last ten generations the predictions improved significantly: FDC explained 73 % and 58 % of the variations in the optimization speed of the test functions. FDC also explains some variations in the optimal population size, a fact that may be useful in practice. FDC measures only linear correlation between fitness and Hamming distance to the solution. In many cases, with strongly nonlinear fitness landscape, other correlation metrics are expected to be more appropriate. Fitness distance analysis could perhaps be useful in development of new genetic operators that were robust to translations of the optima.

Acknowledgements

The author would like to thank the Research Institute for Technology at the University of Vaasa for financial support.

References

J.T. Alander. An Indexed Bibliography of Genetic Algorithms: Theory and Comparisons. University of Vaasa, 1994. URL: ftp://ftp.uwasa.fi/cs/report94-1.

J.T. Alander. Potentials of Genetic Algorithms. TEKES, 2002. URL: ftp://ftp.uwasa.fi/cs/report96-1/English.ps.Z.

L. Altenberg. Fitness distance correlation analysis: an instructive counterexample. In: *Proceedings of the Seventh International Conference on Genetic Algorithms* (ICGA97), 1997.

T. Bäck, D.B. Fogel, Z. Michalewicz (eds.). *Evolutionary Computation* 2, 2002. ISBN: 0750306653.

Y. Borenstein, R. Poli. Fitness distributions and GA hardness. In: *Proceedings of PPSN04*, 2004.

S. Forrest, M. Mitchell. Relative building block fitness and the building-block hypothesis. In: D. Whitley (ed.) *Foundations of Genetic Algorithms* 2, Morgan Kaufmann, San Mateo, 1993.

D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Morgan Kaufmann, 1989.

P. Hartono, S. Hashimoto S., M. Wahde. Labeled-GA with adaptive mutation rate. In: *Proc. IEEE CECI*, 2004.

T. Jones, S. Forrest. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In: Larry Eshelman (ed.) *Proceedings of the* 6*th* *International Conference on Genetic Algorithms*, San Francisco, CA, 1995.

R.A. Mansfield. *Genetic Algorithms*, University of Wales, 1990.

E. Popovici, K. De Jong. Understanding EA dynamics via population fitness distributions. In: Alwyn Barry (ed.) *Workshop Program Proceedings of Genetic and Evolutionary Computation Conference* (GECCO03), 2003.

.S. Rana. Examining the role of local optima and schema processing in genetic search. PhD thesis, Colorado State University, 1998.

G. Syswerda. Uniform crossover in genetic algorithms. In: *Proc. of the Third International Conference on Genetic Algorithms*, 1989.

L.Vanneschi, M. Tomassini. A study on fitness distance correlation and program difficulty for genetic programming. In: *Workshop Program Proceedings of Genetic and Evolutionary Computation Conference,* (GECCO02), 2002.

L. Vanneschi, M. Tomassini. Pros and cons of fitness distance correlation in genetic programming. In: Alwyn Barry (ed.) *Workshop Program Proceedings of Genetic and Evolutionary Computation Conference* (GECCO03), 2003.

Building Video Databases to Boost Performance Quantification – The DXM2VTS Database

Dereje Teferi[⋆], Josef Bigun[⋆]
[⋆]School of Information Science, Computer, and Electrical Engineering (IDE)
Halmstad University
P.O.Box 823, SE-301 18
Halmstad, Sweden
{Dereje.Teferi,Josef.Bigun}@ide.hh.se

Abstract

Building a biometric database is an expensive task which requires high level of cooperation from a large number of participants. Currently, despite increased demand for large multimodal databases, there are only a few available. The XM2VTS database is one of the most utilized audio-video databases in the research community although it has been increasingly revealed that it cannot quantify performance of a recognition system in the presence of complex background, illumination, and scale variability. However, producing such databases could mean repeatedly recording a multitude of audio-video data outdoors, which makes it a very difficult task if not an impossible one. This is mainly due to the additional demands put on participants. This work presents a novel approach to audio-visual database collection and maintenance to boost the performance quantification of recognition methods and to increase the efficiency of multimodal database construction. To this end we present our segmentation procedure to separate the background of a high-quality video recorded under controlled studio conditions with the purpose to replace it with an arbitrary complex background. Furthermore, we present how an affine transformation and synthetic noise can be incorporated into the production of the new database to simulate real noise, e.g. motion blur due to translation, zooming and rotation. The entire system is applied to the XM2VTS database, which already consists of several terabytes of data, to produce the DXM2VTS – Damascened XM2VTS database essentially without an increase in resource consumption, i.e. storage space, video operator time, and time of clients populating the database. As a result, the DXM2VTS database is a damascened (sewn together) composition of two independently recorded real image sequences that consist of a choice of complex background scenes and the the original XM2VTS database.

1 Introduction

Biometrics is an important and increasingly challenging research topic. It is the automatic recognition of individuals based on who they are (e.g. face, iris, fingerprint etc) and/or what they can do (e.g. voice, signature) instead of what they hold (e.g. ID cards) and/or what they remember (e.g. PIN, password) (Ortega-Garcia et al., 2004). Biometric data such as face, fingerprints, iris, ear, hand, and voice are used to track and analyze features and uniquely identify and verify people. Such systems are built through years of research, testing, experimentation and innovation. Their performance quantification depends on, among other things, the size and variety of the database used.

Biometrics technology gives a high level of authentication and is successful in many ways but has not yet been fully trusted by users due to fraud and impersonation. Even so, they have become an integral part of infrastructure used for diverse business sectors such as security, finance, health, law enforcement etc (Jain et al., 2004a), (Bailly-Bailliére et al., 2003). Practical biometric systems should be accurate to a specified level, fast, harmless to users, and accepted by users as well (Ortega-Garcia et al., 2004), (Jain et al., 2004b). User acceptance and high confidence can be achieved by training, testing and evaluating these biometric systems on variety of large databases recorded in real world environment, using multiple modalities for authentication, and incorporating aliveness detection into the systems.

One of the major problems in the development of biometric systems is the lack of public large databases acquired under real-world environment for training, testing and evaluation. Those systems trained and tested on databases recorded in realistic working and living environments perform evidently better. However, most biometric databases are built in a studio and have near-constant color background such as XM2VTS (blue), CUAVE (green) etc. Although one color background does not represent real life scenery, it facilitates segmentation of the background to replace it with a realistic one. However, this has not been done before for a variety of reasons. Whereas the background is illuminated uniformly in CUAVE enabling chroma-keying of different backgrounds (Patterson et al., 2002), this is not the case in XM2VTS as the background is not uniformly blue, Fig.1.

The need to collect new databases, consuming huge resources and claiming important maintenance resources, to represent real world situations is stated in (Bailly-Bailliére et al., 2003) as:

> *The XM2VTS database, together with the Lausanne protocol contains 295 subjects recorded over 4 sessions. However, it was not possible to use it as the controlled recording environment was not realistic enough compared to the real world situations when one makes a transaction at home through a consumer web cam or through an ATM in a variety of surroundings. Therefore it was decided that a new database for the project would be recorded and a new experimental protocol using the database defined.*

Recording a database in real world (non-controlled) environments representing multiple scenarios is difficult if not impossible. This is due to the high demand it puts on the participants to repeatedly appear on weekly or monthly intervals to real-life environments (e.g. malls, busy stations, streets, etc) for recording. To this end, we propose a method to segment the image sequences of the XM2VTS database and create a binary mask of the background and the foreground. These masks will be used to generate various damascened XM2VTS databases with realistic background, such as shopping centers, moving cars, people, additionally, they will be used to add scale variation, illumination, motion and zooming blur to improve performance quantification of biometric systems.

This paper is organized as follows. The next two sections give some background on the XM2VTS database and its specific demands on image segmentation. In section four and its subsections we present our approach. The subsections include the three stages of the proposed system: segmentation, compression of the binary mask, and building the damascened database. In section five we present the experiment conducted and finally our conclusions are summarized.

2 The XM2VTS Database

The XM2VTS database is a 295 subject audio-video database that offers synchronized video and speech data as well as side view images of the subjects. Its design is based on the experience of its predecessor, the M2VTS database which is comprised of 37 subjects. The database acquisition started with 360 subject and 295 of them completed the whole session of four recordings. These subjects are recorded inside a studio in four separate sessions in a uniformly distributed period of five months. This ensures the natural variability of styles and moods of subjects over different times. The database is intended for researchers in areas related to multimodal recognition systems (Messer et al., 1999).

2.1 Performance quantification

XM2VTS has associated evaluation protocols to assess the performance of vision and speech based systems developed using this database as a training and test bed. It has been used in the published studies of many researches and therefore it is considered as a useful comparison yard-stick (Bailly-Bailliére et al., 2003). The results from two face verification contests on the XM2VTS database that were organized in conjunction with the 2000 International Conference on Pattern Recognition and the AVBPA 2003 are reported in (Matas et al., 2000) and (Messer et al., 2003) respectively.

However, XM2VTS and many other face image databases are designed mainly to measure performance of face recognition methods and thus the images contain only one face. Near-constant background one-face images does not require the system to track the faces from complex backgrounds. They are suitable for training recognition systems rather than testing as the tacit reason for comparing classifiers on test sets is that these data sets represent problems that systems might face in the real world. Moreover, it is assumed that superior performance on these benchmarks may translate to superior performance on other real-world tasks (Yang et al., 2002). To this effect, it is important to boost the performance quantification of recognition systems trained on the XM2VTS database by synthesizing the image sequences with a real-world background and a simulation of natural noise.

Accordingly, research in biometrics need many databases similar to that of XM2VTS, with varying background, to build safer and better systems for the ever increasing security requirements of the public at large.

3 Image Segmentation

Image segmentation is used and applied in many computer vision research and applications. It is a problem for which no general solution exist. It is broadly defined as partitioning of pixels into groups with similar properties.

Generally color (in different color spaces), edge, texture and relative spatial location are the factors that define the properties of an image. These factors have been used to develop many algorithms for image segmentation. A survey of many image segmentation techniques is discussed in (Cheng et al., 2001) and (Lucchese and Mitra, 2001). Although many of the algorithms developed do not require training, some application specific knowledge needs to be provided as a parameter, typically a threshold on within-class variance or the number of classes.

There exists a multitude of clustering techniques such as hierarchical clustering, kmeans, and fuzzy c-means that are standard in many mathematical software packages. However, they need to be adapted to the needs of automatic image segmentation e.g. the spatial continuity of the assigned classes are not guaranteed which in turn requires post processing, or when thousands of images are individually segmented the number of classes parameter that is needed as input may vary, which in turn prompts for manual intervention. Below, we summarize the various stages and the adaptation that we have undertaken to minimize the manual intervention during the automatic segmentation of hundreds of thousands of images included in the XM2VTS database.

4 Segmentation, Adding Variability and Building the New Database

4.1 Segmentation

Histogram thresholding method is tested to see if we can separate the background of image sequences from the XM2VTS video database. However, finding a single threshold group that works for hundreds of thousands of images can not be achieved due to varying illumination of the background within an image as well as from image sequence to image sequence. As a result, the segmentation showed poor performance, Fig. 1.

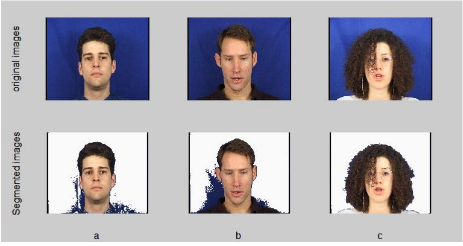


Figure 1: Segmentation by histogram thresholding

Here we present the technique we used to separate a near-constant color background from a face-shoulder image of a video. We applied it on the XM2VTS database to create the necessary mask for the rest of the application. The segmentation technique is an iterative procedure that uses two feature spaces. The first iteration creates a crude set of classes and then a refinement process follows that iteratively adjusts the class assignments and their centroid. This procedure is presented below.

4.1.1 Clustering

A low-pass filtering using a Gaussian filter is applied to smooth the image before segmentation to increase the signal to noise ratio. The Gaussian filter G is a separable filter and has the parameter σ:

$$G = exp(-(x^2 + y^2)/2\sigma^2) \qquad (1)$$

The segmentation algorithm is applied on the smoothed image. The proposed segmentation does not require *a priori* knowledge of the number of clusters in the image as this is automatically determinable as will be explained below. Automatic determination of class numbers is necessary in our work as we have thousands of image sequences to segment and the number of clusters may vary from image sequence to image sequence depending on the color and texture of the subjects' clothing.

4.1.2 The feature space and the clustering criteria

The segmentation process needs a *feature space*, which is normally a vector space, equipped with a

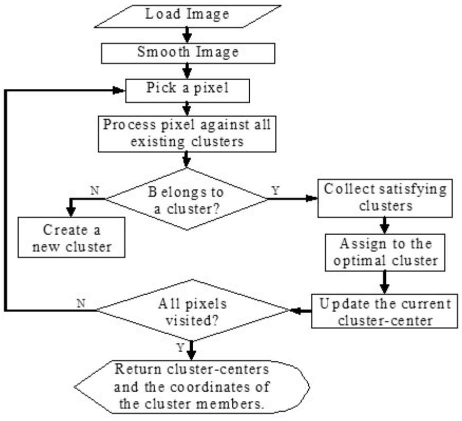

Load Image
Smooth Image
Pick a pixel
Process pixel against all existing clusters
Belongs to a cluster?
N
Y
Create a new cluster
Collect satisfying clusters
Assign to the optimal cluster
Update the current cluster-center
All pixels visited?
N
Y
Return cluster-centers and the coordinates of the cluster members.

Figure 2: The segmentation algorithm

metrics to measure distances. We use here two such vector spaces jointly. The first one consists of the ordinary rgb-color space equipped with the max-norm. To be precise, let $\mathbf{f}_{ij}$ represent the color vector of a pixel at row i, column j of an image then

$$\mathbf{f}_{ij} = (f^1_{ij}, f^2_{ij}, f^3_{ij})^T \tag{2}$$

where the superscripts 1, 2, and 3 represent the color components red, green, and blue.

The max-norm of a vector $\mathbf{f}$ is then defined as,

$$\|\mathbf{f}\|_\infty = \|(f^1, f^2, f^3)^T\|_\infty = \max_k\{f^k\} \tag{3}$$

where $\quad k \in \{1, 2, 3\}$

Having the same metrics as above, the second feature vector space is constructed from the first one as follows:

$$\begin{aligned}\tilde{\mathbf{f}}_{ij} &= (\tilde{f}^1_{ij}, \tilde{f}^2_{ij}, \tilde{f}^3_{ij})^T \\ &= (f^1_{ij} - f^2_{ij}, f^2_{ij} - f^3_{ij}, f^3_{ij} - f^1_{ij})^T \end{aligned} \tag{4}$$

This feature space is introduced to split regions that have too high hue variations since it more directly measures the direction of an rgb-color vector, the hue, than the first feature space.

To describe the clustering algorithm, we need to define *the distance to cluster-center vector* of a pixel at row i, column j as

$$\delta^l_{ij} = (\|\mathbf{f}_{ij} - \mathbf{c}^l\|_\infty, \|\tilde{\mathbf{f}}_{ij} - \tilde{\mathbf{c}}^l\|_\infty)^T \tag{5}$$

where $\mathbf{c}^l$ is the cluster-center vector of the cluster with label l and $\mathbf{f}_{ij}$ represents the rgb-color vector of a pixel. The vectors $\tilde{\mathbf{c}}^l$, $\tilde{\mathbf{f}}_{ij}$ are the cluster-center vector, and the feature vector of the pixel at row i, column j but measured in the second feature space.

The clustering and segmentation is achieved by iteratively updating the partitioning, and the cluster-centers, for every advancement of the pixel position, Fig. 2. The pixel position advancement follows the scan-direction, which attempts to achieve a spatial continuity of the labels being obtained in the segmentation process. A pixel is considered to belong to a cluster, if its distance to cluster-center vector is within certain limits,

$$0 \le \delta^l_{ij} \le \tau$$

where $\tau = (\tau^1, \tau^2)^T$ represents a threshold vector, which determines the maximum variation of the distance to cluster-center vector allowed within the same cluster, in each feature space independently. It is worth noting that in traditional clustering algorithms used in image segmentation, the pixel positions have no-influence on cluster-assignments. Yet, since the physics of imaging is continuous w.r.t. pixel positions, the class-labels should change smoothly, prompting for an additional step attempting to remove "salt-pepper" classes scattered around large classes.

The procedure has been implemented by visiting pixels in three different (sequential(scan-line), random, and hilbert curve) directions. For the XM2VTS database, we have verified that the segmentation responded better for the hilbert and the scan-line traversals. This is mainly attributed to the continuity of intensity changes in the background of the images. The sequential visiting order is chosen for efficiency reasons only. This order of presenting pixels to the clustering scheme attempts to enforce continuity at least line-wise.

A pixel's distance to all available cluster-centers are computed and the pixel is assigned to the "closest" cluster, the nearest neighbor assignment. To give a meaning to "closeness", in the combined feature spaces, the metrics of the two feature spaces are merged in a straightforward manner as

$$\|\delta^l_{ij}\|_\infty = \|(\|\mathbf{f}_{ij} - \mathbf{c}^l\|_\infty, \|\tilde{\mathbf{f}}_{ij} - \tilde{\mathbf{c}}^l\|_\infty)^T\|_\infty \tag{6}$$

Finally a cluster-list containing all the clusters and their centroids is generated from this procedure, Table 1. This is used as an initial cluster for the cluster refinement procedure described next.

Table 1: Cluster matrix

Cl	Red	Green	Blue	N_pxls	Members
1	0.153	0.133	0.134	4837	List_1
2	0.159	0.221	0.518	37341	List_2
3	0.186	0.259	0.614	27735	List_3
4	0.120	0.153	0.293	1688	List_4
.	.	.	.	.	.
.	.	.	.	.	.

4.1.3 Cluster refinement

Cluster refinement is a procedure applied on the image to refine the cluster-centers as well as the their member pixels. This procedure is necessary in our method as the segmentation above terminates upon visiting each pixel once. We use the same procedure as above when refining the clusters except that the cluster matrix we obtained from the initial segmentation is used as the start cluster list for the refinement. The segmentation result converges by running this procedure repeatedly while automatically adjusting the class centroids as well as the number of classes over each iteration.

Once the clusters are refined, the background class is identified as the the cluster with the largest member pixels. We verified that this hypothesis indeed holds very well for the XM2VTS database, although more refined approaches, such as the cluster containing most boundary pixels, and the cluster-center closest to a certain set of example background colors, can be used either separately or in combination. Finally, a 2D mask is created by identifying the background cluster pixels and assigning them the logical value one whereas the members of the remaining clusters (the foreground) are set to zero.

4.1.4 Improving the efficiency of segmentation

The speaking head shots of the XM2VTS database comprise about 75% of the overall image sequences. They are used for speech and/or face-recognition applications. These shots contain only small amount of motion of the person as a natural behaviour of the subjects when speaking. We exploited this feature to improve the efficiency of our segmentation algorithm for the speaking head shots of the XM2VTS database. That is, the segmentation result of the first image can be used effectively for the segmentation of the remaining images within an image sequence.

The binary mask we obtained from the segmentation of the first frame is used to create a narrow band around the border of the foreground image part, which we below refer to as *the narrow band*. In the consecutive frames of the image sequence only the image part corresponding to the pixel coordinates of the narrow band, from the binary mask of the first frame, is passed through the segmentation scheme presented above.

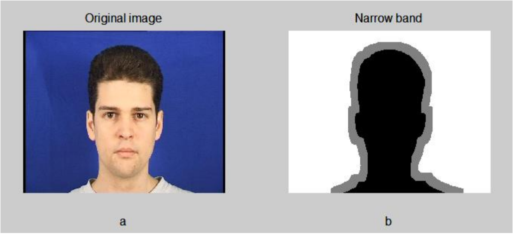


Figure 3: Narrow band around the foreground

The motion of the subject is only within the marked narrow band, as shown in Fig. 3. Therefore we know that the zero, or the black label, represents the foreground whereas the white part is the background in all frames of the video. Accordingly, the segmentation process for the consecutive frames is applied on the narrow band and the part that belongs to the background is set to one and the rest to zero. This result is subsequently used as a binary image mask for the specific frame in the image sequence.

This procedure could not be applied for the rotating heads image sequences as the motion of the subjects cover almost all areas of the frame. Thus, for the case of the rotating-head shots, the segmentation procedure is applied on the whole image to create the binary mask of each frame.

4.2 Compression of the binary mask

Each binary mask of a video is basically an *fr* dimensional binary image of size (r,c,fr), where (r,c) is the size of an image in the sequence and fr is the number of frames in the video. Storing these binary masks of every frame of the image sequence is possible but not efficient especially for distribution as the size of video databases, such as that of XM2VTS, is very large. Thus, compression is found to be necessary. There are several compression algorithms available. The test results of MPEG-4 and RLE – Run Length Encoding are presented below.

4.2.1 Compression using MPEG-4 encoding

MPEG-4 is an ISO/IEC standard developed by MPEG (Moving Picture Experts Group) in 1999. The fully backward compatible extensions under the title of MPEG-4 Version 2 became an international standard in 2000[1].

The binary mask of each image sequence is converted to an Audio/Video Interleaved (AVI) file and compressed[2]. The compression level (lossy) at a data rate of 512 Kb/sec is only 10% whereas at a data rate of 256 Kb/sec is about 54%. However, the compression is not only lossy but also visually degraded on the latter case and therefore not convenient for our purpose.

4.2.2 Compression using RLE

Run Length Encoding (RLE) is a process of searching for repeated runs of a symbol in an input stream and replacing them by a single instance of the symbol and its run count. The repetitive appearance of 1's and 0's in the binary masks makes it convenient for RLE compression. Moreover, the binary mask, in our case contains only two values, 1's and 0's, and therefore modifying the RLE algorithm by storing the first symbol once and then only the run count of the consecutive symbols is possible. That is, we only store the run count of the symbols and not the symbols themselves, Table 2. This again improves the RLE algorithm by reducing the size of the compressed mask substantially. Furthermore, we use a zero delimiter as a separator between frame masks.

Table 2: RLE vector

First symbol	count of first symbol	count of next symbol	.	.	end of frame	.
1	37234	35	.	.	0	.

The size of the RLE vector is 70 – 80% less than the size of the original binary mask. Moreover, it is easy to store, distribute and process with less memory requirement.

Therefore, the modified RLE algorithm is implemented in this system as it performed better for the specific case of compressing the binary masks of a face-shoulder video database.

4.3 Building the damascened database

The next step is building the damascened XM2VTS image sequence from the compressed mask. The mask is decompressed using a reverse-algorithm of the one that is used for compression. This generates the narrow band matrix. Then, we use each row of the narrow band matrix as the values of the narrow band of a binary image mask to generate the binary mask of the whole image sequence.

Parallel frames are extracted from the mask, its corresponding XM2VTS video, and the new background video, one at a time to build the damascened database. In practice, sewing together, or damascening, the two real image sequences according to the mask, can be easily achieved by multiplying the binary mask with the background, its inverse with the XM2VTS frame and adding the result to get the required frame of the synthetic image sequence. At this point, distortion is added as required on the synthetic frame to simulate real recording environments, Fig. 4. Real videos can be distorted or blurred for a variety of reasons. For example, while recording objects moving with high speed, the camera must often be moved manually to keep the moving object in the center. When tracking a still object when the camera is moving, e.g. in a car, high speed motion blur is generated for the rest of the objects that are not tracked. Noise could also occur from poor quality of camera used or due to weather conditions at the time of recording.

To simulate a significant portion of the natural noise, we suggest to apply distortions to the synthetic image sequence according to the following models:

- Horizontal blur: translation of pixels in the x directions;
- Vertical blur: translation of pixels in the y directions;
- Zoom blur: translation of pixels in the radial direction;
- General motion blur: affine transformation
- Imaging noise: Salt and pepper noise at different noise intensity levels.

A set of standard video backgrounds such as offices, outdoors, malls, moving cars is collected and offered with the compressed mask of the XM2VTS database for distribution. These data can be used to generate damascened XM2VTS database with required level of noise and the variability of the background due to the real life scenario changes.

5 Experimental results

A series of experiments are conducted on 55 randomly selected XM2VTS database sequences to de-

[1] The MPEG Home Page. http://www.chiariglione.org/mpeg/

[2] Compressed using MPEG-4 encoding by an open source software virtualDub, http://www.virtualdub.org/

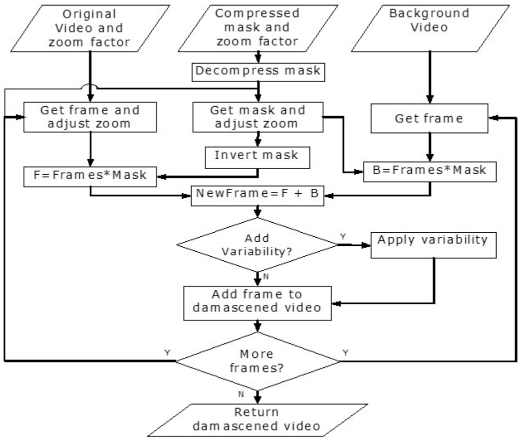


Figure 4: Algorithm for building the new video

termine the optimal threshold for distance to cluster-centers and intensity variations, τ_1, τ_2 where the background would be separated as one cluster. These values are empirically found to be 0.35 and 0.23, assuming an image where the rgb-color components of pixels are between 0 and 1. In addition, the motion of the subjects while speaking is found to be within a 24 pixel wide narrow band. The Gaussian filter used to smooth the image frames before segmentation is set to be of size (7x7) and $\sigma = 2.5$. Moreover, only two iterations are found to be necessary to separate the background of the XM2VTS images.

Using these parameters, 94% of the image sequences of the database are segmented successfully. Since the segmentation of consecutive images is done only on the narrowband that is collected from the first frame of the image sequence, the remaining 6% of the image sequences are segmented semi-automatically by human intervention only on the first frame to find the narrow band.

A software to generate synthetic XM2VTS database from the binary masks is built using Matlab 7.1. A set of representative distortion parameters such as salt and pepper noise, blurring and affine transformation are included on the software to simulate motion of camera, rotation, blurring etc in the output synthetic video as shown in Fig. 5.

6 Conclusion

The segmentation algorithm in our system uses a variant of the max-norm applied to two color spaces simultaneously. These metrics are combined together to yield a segmentation accuracy of 94% at the level of image sequences, i.e. when checked visually, the boundaries found appeared natural to a human observer (the authors).

Due to its public availability, currently, the XM2VTS database is one of the largest, and probably the most utilized biometric multimodal (face and audio) databases in existence, at least in academic research. A method is presented in this paper to store a binary mask for the speaking image sequences of this database. The result which contains a collection of compressed masks is suggested to be used to sew together the XM2VTS database head-shoulders with complex image sequences to obtain damascened image sequences. The boundaries have a high accuracy whereas the damascened sequence contains realistic, yet controllable distortions, such as the amount of the motion blur.

Acknowledgment

This work has been sponsored by the Swedish International Development Agency (SIDA).

References

E. Bailly-Baillière, S. Bengio, F. Bimbot, M. Hamouz, J. Kittler, J. Mariéthoz, J. Matas, K. Messor, V. Popovici, F. Porée, B. Ruiz, and J. Thiran. The BANCA database and evaluation protocol. In *Audio and Video Based Person Authentication: AVBPA 2003, LNCS 2688*, pages 625–638, 2003.

H.D. Cheng, X.H. Jiang, and J. Wang. Colour image segmentation: Advances and prospects. *Pattern Recognition Letters*, 34:1277–1294, 2001.

A.K. Jain, S. Pankanti, S. Prabhakar, L. Hong, and A. Ross. Biometrics: A grand challenge. In *Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004*, volume 2, pages 935–942, 2004a.

A.K. Jain, A. Ross, and S. Prebhakar. An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on Image- and Video-Based Biometrics*, 14(1), January 2004b.

L. Lucchese and S.K. Mitra. Colour image segmentation: A state-of-the-art survey. In *Image Processing, Vision, and Pattern Recognition, Proc. of the Indian National Science Academy (INSA-A)*, volume 67A(2), pages 207–221, March 2001.

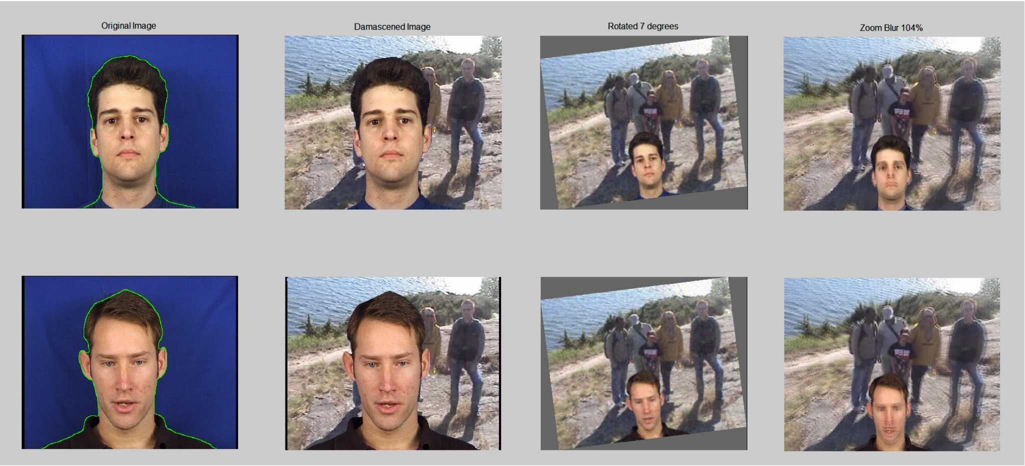


Figure 5: Image frames from the damascened video

J. Matas, M. Hamouz, K. Jonsson, J. Kittler, Y. Li, C. Kotropoulos, A. Tefas, I. Patas, T. Tan, H. Yan, F. Smeraldi, J. Bigun, N. Capdevielle, W. Gerstner, S. Ben-Yacoub, Y. Abdeljaoued, and E. Mayoraz. Comparison of face verification results on the XM2VTS database. In *Proceedings of International Conference on Pattern Recognition. ICPR-15*. IEEE Computer Society, 2000.

K. Messer, J. Matas, J. Kitler, J. Luettin, and G. Maitre. XM2VTSDB: The extended M2VTS database. In *2nd International Conference on Audio and Video-based Biometric Person Authentication (AVBPA'99)*, pages 72–77, 1999.

K. Messer, J. Kittler, M. Sadeghi, S. Marcel, C. Marcel, S. Bengio, F. Cardinaux, C. Sanderson, J. Czyz, L. Vandendorpe, S. Srisuk, M. Petrou, W. Kurutach, A. Kadyrov, R. Paredes, B. Kepenekci, F.B. Tek, G.B. Akar, F. Deravi, and N. Mavity. Face verification competition on the XM2VTS database, 2003.

J. Ortega-Garcia, J. Bigun, D. Reynolds, and J. Gonzalez-Rodriguez. Authentication gets personal with biometrics. *IEEE Signal Processing Megazine*, 21(2):50–62, March 2004.

E.K. Patterson, S. Gurbuz, Z. Tufekci, and J.N. Gowdy. CUAVE: A new audio-visual database for multimodal human-computer interface research. In *International Conference on Acoustics, Speech, and Signal Processing, Proceedings. (ICASSP 2002)*, pages 2017–2020, May 2002.

M-H. Yang, D.J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligece*, 24(1): 34–58, January 2002.

Fast n-Fold Cross-Validation for Regularized Least-Squares

Tapio Pahikkala[⋆] Jorma Boberg[⋆]
Tapio Salakoski[⋆]
[⋆]Turku Centre for Computer Science (TUCS), Department of Information Technology, University of Turku
Lemminkäisenkatu 14 A, FIN-20520 Turku, Finland
firstname.lastname@it.utu.fi

Abstract

Kernel-based learning algorithms have recently become the state-of-the-art machine learning methods of which the support vector machines are the most popular ones. Regularized least-squares (RLS), another kernel-based learning algorithm that is also known as the least-squares support vector machine, is shown to have a performance comparable to that of the support vector machines in several machine learning tasks. In small scale problems, RLS have several computational advantages as compared to the support vector machines. Firstly, it is possible to calculate the cross-validation (CV) performance of RLS on the training data without retraining in each CV round. We give a formal proof for this claim. Secondly, we can compute the RLS solution for several different values of the regularization parameter in parallel. Finally, several problems on the same data set can be solved in parallel provided that the same kernel function is used with each problem. We consider a simple implementation of the RLS algorithm for the small scale machine learning problems that takes advantage of all the above properties. The implementation is done via the eigen decomposition of the kernel matrix. The proposed CV method for RLS is a generalization of the fast leave-one-out cross-validation (LOOCV) method for RLS which is widely known in the literature. For some tasks, the LOOCV gives a poor performance estimate for the learning machines, because of the dependencies between the training data points. We demonstrate this by experimentally comparing the performance estimates given by LOOCV and CV in a ranking task of dependency parses generated from biomedical texts.

1 Introduction

Kernel-based learning algorithms (Schölkopf and Smola, 2002; Shawe-Taylor and Cristianini, 2004) have recently become the state-of-the art machine learning methods of which the support vector machines are the most popular ones. In this paper, we consider Regularized least-squares (RLS) algorithm (see e.g. Rifkin (2002); Poggio and Smale (2003)), another kernel-based learning algorithm that is also known as the least-squares support vector machine (Suykens and Vandewalle, 1999). They were shown to have a performance comparable to that of the support vector machines in several machine learning tasks. Traditionally, RLS type of algorithms have been applied to regression problems but lately they have also been used on other machine learning problems. Recent classification tasks in which RLS have been successfully applied are, for example, disambiguation problems in natural language (Popescu, 2004), DNA classification (Ancona et al., 2005), and classification of intensive care nursing narratives (Hiissa et al., 2006). Another successful application area has been ranking or ordinal regression (Tsivtsivadze et al., 2005, 2006; Suominen et al., 2006; Pahikkala et al., 2006c).

Cross-validation (CV) is a commonly used method for the performance estimation and model selection for the learning algorithms. For RLS, it is widely known that the leave-one-out cross-validation (LOOCV) has a closed form whose computational complexity is quadratic with respect to the number of training examples. In reality, however, the training set may contain data points that are highly dependent with each other. In text categorization tasks, for example, it may happen that the training set contains several documents written by the same author while that author may not have written the new examples to be predicted with the learning machine. In this kind of situation, the LOOCV performance estimate becomes unreliable, because it may be much easier to predict the labels of the documents when the machine is trained with documents written by the same author. Fortunately, we often have a priori knowledge of such clustering in the training data set and we can perform the CV so that we leave out the whole cluster of data points in each CV round. In this paper, we show that also the leave-cluster-out cross-validation can be performed for RLS with much smaller computational complexity than the naive approach in which the RLS would be retrained in each CV round. To our knowledge, this has not been considered in the literature.

2 Regularization Framework

Let $S = \{(x_1, y_1), \ldots, (x_m, y_m)\} \in (X \times \mathbb{R})^m$ be a training set of m training examples, where $x_i \in X$ are the training data points, $y_i \in \mathbb{R}$ are their labels, and X can be any set. We consider the Regularized Least-Squares (RLS) algorithm as a special case of the following regularization problem known as Tikhonov regularization (for a more comprehensive introduction, see e.g. Poggio and Smale (2003)):

$$\min_f \sum_{i=1}^{m} l(f(x_i), y_i) + \lambda \|f\|_k^2, \tag{1}$$

where l is the loss function used by the learning machine, $f : X \to Y$ is a function which maps the inputs $x \in X$ to the outputs $y \in Y$, $\lambda \in \mathbb{R}_+$ is a regularization parameter, and $\|\cdot\|_k$ is a norm in a Reproducing Kernel Hilbert Space defined by a positive definite kernel function k. The second term is called a regularizer. The loss function used with RLS for regression problems is called least squares loss and is defined as

$$l(f(x), y) = (y - f(x))^2.$$

Note that if we use $l(f(x), y) = \max(y - f(x), 0)$, we obtain the support vector machines (SVM) for classification. Other choices of the loss function lead to other popular classifiers, for example, the SVM regression and kernel logistic regression. By the Representer Theorem (see e.g. Schölkopf et al. (2001)), the minimizer of equation (1) with the least-squares loss function has the following form:

$$f(x) = \sum_{i=1}^{m} a_i k(x, x_i), \tag{2}$$

where $a_i \in \mathbb{R}$ and k is a kernel function.

3 Implementation

We now state the solution of RLS in the case where there are several output labels for each data point (see e.g. Rifkin and Klautau (2004) for a more comprehensive consideration). Below, $\mathcal{M}_{i \times j}(\mathbb{R})$ denotes the set of real valued matrices of dimension $i \times j$. Suppose we have a set of m training examples $S = \{(x_1, y_1), \ldots, (x_m, y_m)\} \in (X \times \mathbb{R}^p)^m$, where $x_i \in X$ are the input variables, X can be any set, and y_i are the corresponding vectors of p output variables. Let $K \in \mathcal{M}_{m \times m}(\mathbb{R})$ be the kernel matrix generated from the training data points using the kernel function $k(x, z)$, that is, $K_{ij} = k(x_i, x_j)$. Let $Y \in \mathcal{M}_{m \times p}(\mathbb{R})$ be a matrix whose rows are the vectors of the output variables, that is, it has one column per each subproblem. Further, let $G = (K + \lambda I)^{-1}$, where $I \in \mathcal{M}_{m \times m}(\mathbb{R})$ is the identity matrix. Because the kernel function from which the kernel matrix is generated is positive definite, the matrix $K + \lambda I$ is invertible when $\lambda > 0$. Given a regularization parameter λ, the coefficient matrix is obtained as follows

$$A = GY \in \mathcal{M}_{m \times p}(\mathbb{R}) \tag{3}$$

whose columns are the coefficient vectors of the RLS solution (2) for each p output (for a proof, see e.g. Rifkin (2002)). Because the kernel matrix K is the same for all of the p problems, we obtain all solutions from (3) with approximately the cost of solving only one problem.

3.1 Solving RLS via eigen decomposition of kernel matrix

Solution (3) can be obtained simply by calculating the inverse G of the matrix $K + \lambda I$. Instead of calculating the inverse, we compute the solution by first calculating the eigen decomposition of the kernel matrix

$$K = V \Lambda V^{\mathrm{T}}, \tag{4}$$

where V is an orthogonal matrix that contains the eigenvectors of K and Λ is a diagonal matrix that contains the corresponding eigenvalues. Then, $G = (V \Lambda V^{\mathrm{T}} + \lambda I)^{-1} = V \widetilde{\Lambda}_\lambda V^{\mathrm{T}}$, where $\widetilde{\Lambda}_\lambda = (\Lambda + \lambda I)^{-1}$ is a diagonal matrix that contains the eigenvalues of G. The ith eigenvalue of G is $1/(\mu_i + \lambda)$, where μ_i is the ith eigenvalue of K. Note that we do not need to compute the matrix G, because the solution (3) can now be obtained as

$$A = V \widetilde{\Lambda}_\lambda V^{\mathrm{T}} Y. \tag{5}$$

From (5) we observe that after we have calculated the eigen decomposition of K, we can easily compute a whole array of RLS solutions for different values of λ. To compute the solution (5) for a certain value of λ, we need to calculate the product of the matrices

$V\widetilde{\Lambda}_\lambda \in \mathcal{M}_{m\times m}(\mathbb{R})$ and $V^{\mathrm{T}}Y \in \mathcal{M}_{m\times p}(\mathbb{R})$ which is fast when the number of subproblems p is small compared to the number of training examples m. We can then select the regularization parameter for each subproblem with a cross-validation which we consider below. Of course, different subproblems may prefer different values of λ. In that case, we do not obtain the column vectors of A from (5), but one by one from $A_p = V\widetilde{\Lambda}_{\lambda_p}V^{\mathrm{T}}Y$, where a_p is the pth column of A and λ_p is the value of the regularization parameter preferred by the pth subproblem.

3.2 Efficient Computation of Cross-Validation

We now consider an efficient computation of cross-validation (CV) for the RLS algorithm. By CV we indicate the method that is used to estimate the performance of the learning algorithm with a given data set. The outline of the method is the following. First, the data set is partitioned into subsets called CV folds. Next, the learning machine is trained with the whole data set except one of the folds that is used to measure the performance of the machine. Each of the CV folds is held out from the training set at a time and the CV performance of the machine is obtained by averaging over the performances measured with the different folds.

In order to calculate the CV performance of a learning machine explicitly, we need to train the machine n times, where n is the number of the CV folds. This is, in many cases, computationally cumbersome, especially if the number of the folds is large. Fortunately, it is possible to obtain the CV performance of RLS with a smaller computational cost than in the naive approach.

We now prove a lemma that we use below to derive a faster method for the computation of CV. The proof is similar to the proof of the leave-one-out lemma (see e.g. Wahba (1990)). For simplicity, we prove only the case in which the number of output variables is one. However, it is easy to extend the lemma for p output variables.

Lemma 1. *Let L be a set of indices of the data points that are held out from the training set and let f_L be the function obtained by training the RLS algorithm with the whole data set except the set of data points indexed by L. By definition, f_L is the solution to the following variational problem*

$$\min_f \sum_{i\notin L}(f(x_i)-y_i)^2 + \lambda\|f\|_k^2. \tag{6}$$

The function f_L is also the solution to the following variational problem

$$\min_f \sum_{i\notin L}(f(x_i)-y_i)^2 + \sum_{i\in L}(f(x_i)-f_L(x_i))^2 + \lambda\|f\|_k^2$$

Proof. We observe that for any function f

$$\begin{aligned}\sum_{i\notin L}(f(x_i)-y_i)^2 + \sum_{i\in L}(f(x_i)-f_L(x_i))^2 + \lambda\|f\|_k^2 \\ \geq \sum_{i\notin L}(f(x_i)-y_i)^2 + \lambda\|f\|_k^2 \\ \geq \sum_{i\notin L}(f_L(x_i)-y_i)^2 + \lambda\|f_L\|_k^2 \\ = \sum_{i\notin L}(f_L(x_i)-y_i)^2 + \sum_{i\in L}(f_L(x_i)-f_L(x_i))^2 \\ +\lambda\|f_L\|_k^2\end{aligned}$$

□

Using the above lemma, we are able to state the result that allows us to calculate the values of the output variables of the data points held out from the training set without explicitly retraining the algorithm with the rest of the training examples.

Let $I \in \mathcal{M}_{m\times m}(\mathbb{R})$ denote an identity matrix and let $I_L \in \mathcal{M}_{|L|\times m}(\mathbb{R})$ be a matrix that contains the rows of I indexed by L. We use the matrix I_L to "cut" out rows from other matrices, or alternatively to "add" rows consisting of zeros. Below, with any matrix $M \in \mathcal{M}_{m\times n}(\mathbb{R})$, where $n \in \mathbb{N}$, we use the subscript L to denote the left multiplication by I_L, that is, we denote $M_L = I_L M$. Further, we denote $M_{LL} = I_L M I_L^{\mathrm{T}}$ for $M \in \mathcal{M}_{m\times m}(\mathbb{R})$. Let $Y \in \mathcal{M}_{m\times p}(\mathbb{R})$ be the label matrix corresponding to the training data and let us denote $B = KG$. Then

$$B = V\Lambda V^{\mathrm{T}}V\widetilde{\Lambda}_\lambda V^{\mathrm{T}} = V\Lambda\widetilde{\Lambda}_\lambda V^{\mathrm{T}}. \tag{7}$$

By the equation (2), the predicted output of the RLS algorithm for its training data is

$$\widehat{Y} = KA = KGY = BY. \tag{8}$$

Finally, let $Y' \in \mathcal{M}_{m\times p}(\mathbb{R})$ denote the matrix consisting of the output values of the training data points obtained using the function f_L.

Theorem 1. *The matrix Y'_L consisting of the output values of the held out data points predicted with f_L can be obtained from*

$$Y'_L = (I_{LL} - B_{LL})^{-1}(\widehat{Y}_L - B_{LL}Y_L). \tag{9}$$

Proof. According to the Lemma 1, the function f_L is obtained by training the RLS using the whole data set with a label matrix $Y - I_L^{\mathrm{T}} Y_L + I_L^{\mathrm{T}} Y'_L$. Knowing the label matrix, we now use the equation (8) to compute the output matrix Y'.

$$\begin{aligned} Y' &= B(Y - I_L^{\mathrm{T}} Y_L + I_L^{\mathrm{T}} Y'_L) \\ &= \widehat{Y} - B I_L^{\mathrm{T}} Y_L + B I_L^{\mathrm{T}} Y'_L. \end{aligned}$$

By multiplying with I_L from left we get

$$\begin{aligned} Y'_L &= \widehat{Y}_L - B_{LL} Y_L + B_{LL} Y'_L \\ \Leftrightarrow (I_{LL} - B_{LL}) Y'_L &= \widehat{Y}_L - B_{LL} Y_L \\ \Leftrightarrow Y'_L &= (I_{LL} - B_{LL})^{-1} (\widehat{Y}_L - B_{LL} Y_L). \end{aligned}$$

In order to the last equivalence to hold, we have to ensure the invertibility of the matrix $I_{LL} - B_{LL}$. Let γ_i be the ith eigenvalue of B. From (7) we observe that $\gamma_i = (\Lambda \widetilde{\Lambda}_\lambda)_{i,i} = \frac{\mu_i}{\mu_i + \lambda}$. Because $0 \leq \gamma_i < 1$, the matrix $I - B$ is a positive definite. From the positive definiteness of $I - B$, it follows that all its principal submatrices $I_{LL} - B_{LL}$ have strictly positive determinants (see e.g. Meyer (2000)) from which the invertibility follows. □

For a held out set of of size $|L|$, the time consuming part in the computation of (9) is the calculation of the matrix B_{LL}. When we have solved the RLS problem via the eigen decomposition of the kernel matrix, the elements of B_{LL} can be computed using the eigenvectors V and the diagonal elements of $\Lambda \widetilde{\Lambda}_\lambda$, since $B = V \Lambda \widetilde{\Lambda}_\lambda V^{\mathrm{T}}$. Thus, the computational complexity of calculating the outputs Y'_L for the held out set is $O(|L|^2 m)$. If we perform an n-fold cross-validation with the training set, the number and the size of the held out sets are n and m/n, respectively, and the overall complexity of the cross-validation is $O(n(m/n)^2 m) = O(m^3/n)$.

The larger the number of folds in the cross-validation is, the faster is its computation. In the extreme case where the size of the held out set is 1, the computational complexity is $O(m^2)$, since we only have to calculate the diagonal elements of B in the whole cross-validation process. Indeed, from Theorem 1, we obtain as a special case the known result of the leave-one-out cross-validation (LOOCV) for RLS (this case has been proved, for example, by Vapnik (1979); Wahba (1990); Green and Silverman (1994)).

Corollary 1. *Let $f(x_j)$ and $f_j(x_j)$ denote the output of the RLS algorithm for the training example x_j, when the algorithm is trained with all examples and all examples except x_j, respectively. We can calculate the value of the output $f_j(x_j)$ as follows*

$$f_j(x_j) = \frac{f(x_j) - B_{j,j} Y_j}{1 - B_{j,j}}. \tag{10}$$

From the computational complexity perspective, the LOOCV should be preferred with the RLS. However, in practice, there are often cases where the LOOCV should not be used to estimate the performance of the learning algorithm because of dependencies between the training data points. Note also that the sizes of the cross-validation folds do not have to be equal. Therefore, we can divide the training set into folds of different sizes according to the dependencies between the training points. Below, we discuss those cases in more detail.

4 Experiments

With real world data, it is often the case that the data points used to train a machine learning method are not completely independent. For example, we may have several text documents written by the same author in text categorization tasks. A document may be very similar to other the documents written by the same author but very different compared to the documents written by an other author. In these cases, the leave-one-out cross-validation (LOOCV) performance of a learning machine may not be a good estimate of the true performance of the learning machine. This is, because on the contrary to the case with the training set, it is rarely the case that a document to be classified with a trained learning machine is written by the same author as some of the documents in the training set. Generally, we say that the training set consists of clusters of data points. By a cluster we indicate a subset of training examples that are mutually dependent.

Fortunately, we often have a priori knowledge of such clustering in the training data set and we can perform the CV so that we leave out the whole cluster of data points in each CV round. For example, in our earlier experiments using support vector machines and Bayesian classifiers on natural language disambiguation tasks (Pahikkala et al., 2005a,b,c, 2006a,b), we often extracted several examples of the words to be disambiguated from a single text document. When we used bag-of-words kind of features extracted from the whole text, the examples originating from the same text document formed a cluster in the training set. Clearly, it does not happen in practice that a learning machine trained to detect context sensitive spelling errors, is trained with examples originating from the same document the examples to be predicted are originated from. Thus, we had to perform the cross-validation on the document level so that no two examples from the same document end up in different cross-validation folds.

Here we demonstrate the “clustered training set effect” by comparing the LOOCV performance of a trained RLS to a leave-cluster-out cross-validation (LCOCV) performance so that each fold in the LCOCV consists of the training examples that form a cluster in the training set. The demonstration is done with the problem of dependency parse ranking of sentences extracted from biomedical texts. Here we give a brief introduction to the problem, the data, and the solution approach. For a detailed description, see the paper by Tsivtsivadze et al. (2005).

To generate the training data, we took one hundred sentences from the BioInfer corpus (Pyysalo et al., 2006). Each sentence in the corpus has a manually tagged linkage that corresponds to the “correct” parse that a parser is supposed to output for the sentence. For each sentence, we use link grammar parser (Sleator and Temperley, 1991) to generate a set of candidate parses. The number of candidate parses depends of the sentence. If the parser generates more than 20 parses for a sentence, we randomly select 20 of them and discard the rest. Otherwise, we keep all candidate parses. For each candidate parse we calculate a score value that indicates how close to the correct parse it is. The score value is the F-score calculated from the link differences between the candidate parse and the correct parse as follows (Tsivtsivadze et al., 2005):

$$F = \frac{2TP}{2TP + FP + FN},$$

where TP, FP, and FN are the numbers of true positives (the links present in both the candidate and the correct parse), false positives (links present in the candidate parse but not in the correct one), and false negatives (links present in the correct parse but not in the candidate), respectively. The training set is constructed from the candidate parses and their score values. The task of the learning machine is, for a sentence, to rank its candidate parses in the order of their score values. The RLS algorithm is, in fact, trained to regress the score values of the parses but in this paper, we are only interested of the ranking of the candidate parses for each sentence.

We measure the ranking performance by calculating Kendalls τ_b correlation coefficient (Kendall, 1970) for the parse candidate set of each sentence. Note that the coefficient is calculated for each sentence separately, since we are not interested of the mutual order of the parses originating from different sentences. The overall performance for the whole data set is obtained by taking the average of the correlation coefficients of the sentences in the data set.

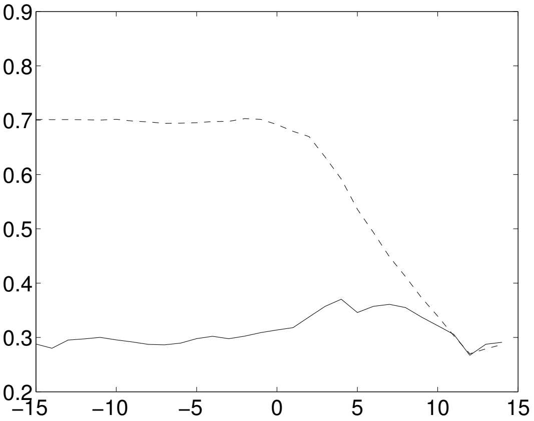


Figure 1: The ranking performance of the RLS algorithm computed with LOOCV (dashed line) and LCOCV (solid line). The x-axis denotes the value of the regularization parameter in a logarithmic scale. The y-axis is the ranking performance measured with τ_b correlation coefficient.

The kernel function, that we use as a similarity measure of the parses, is described in detail by Tsivtsivadze et al. (2005). The kernel function has the drawback that two parses originating from a same sentence have almost always larger mutual similarity than two parses originating from different sentences. Therefore, the data set consisting of the parses is heavily clustered in the feature space determined by the kernel function. The clustered structure of the data can have a strong effect on the performance estimates obtained by cross-validation, because data points that are in the same cluster as a held out point have a dominant effect on the predicted output of the held out point. This does not, however, model the real world, since a parse ranker is usually not trained with parses originating from the sentence from which the new parse with an unknown F-score is originated. The problem can be solved by performing the cross-validation on the sentence level so that all the parses generated from a sentence would always be either in the training set or in the test set.

In order to compare the performance estimates given by the LOOCV and LCOCV, we train an RLS algorithm with the data set described above. Recall that after we have a trained RLS, we obtain the LOOCV output for each parse in the data using (10). From the LOOCV output, we calculate the

F-scores for each sentence and compute their average. We calculate the LCOCV performance by leaving each sentence out from the training set at a time and computing the F-scores for their parses with (9). We make a grid search for the regularization parameter λ of the RLS algorithm with the grid points $2^{-15}, 2^{-14}, \ldots, 2^{14}$. With the grid search, we can test the performances of LOOCV and LCOCV in the model selection of RLS.

The results of the comparison are illustrated in Figure 1. From the figure, we observe that the performance difference between the LOOCV and the LCOCV is, with some values of the regularization parameter, over 0.3 correlation points. Thus, LOOCV clearly overestimates the ranking performance. Moreover, the LOOCV prefers small values of the regularization parameter (the most preferable value of the regularization parameter is 2^{-2}) while the LCOCV prefers a more regularized solution (the most preferable value of the regularization parameter is 2^{4}). Therefore, we can also conclude that the LOOCV may not be a good method for the model selection when the training set is clustered. Indeed, when we test the ranking performance of the RLS with one hundred test sentences unseen to the RLS, we obtain correlations 0.37 and 0.38 with the machines trained with the whole training set and with the regularization parameters preferred by the LOOCV and LCOCV, respectively.

5 Conclusion

The regularized least-squares (RLS) algorithm has been shown to be a competitive alternative to the standard support vector machines in several machine learning tasks. It also have several computational advantages, such as solving several tasks in parallel, fast tuning of the regularization parameter, and fast leave-one-out cross-validation (LOOCV).

The LOOCV method, however, is not always a suitable method for performance estimation or model selection because of dependencies between the training data points. In several learning problems, the training data set may be clustered so that the prediction of a held out data point will be unrealistically easy if the data points that belong in the same cluster with the held out data point are kept in the training set. Since it may not happen in the real world that a data point, whose output is to be predicted with a learning machine, belongs in the same cluster with some of the training data points, the LOOCV estimate does not reflect the reality.

We introduce and prove a closed form of an n-fold cross-validation performance estimate for the RLS algorithm. The closed form can be used to calculate a leave-cluster-out cross-validation (LCOCV), in which a whole cluster of training examples is held out from the training set in each cross-validation round avoiding the harmful effect of the clustered training set.

We experimentally demonstrate the effect of the clustered data set on the LOOCV performance estimate with a ranking task of dependency parses generated from biomedical texts. With the same data and task, it is also demonstrated that the LCOCV is better than LOOCV as a model selection tool.

Acknowledgments

This work has been supported by Tekes, the Finnish Funding Agency for Technology and Innovation.

References

Nicola Ancona, Rosalia Maglietta, Annarita D'Addabbo, Sabino Liuni, and Graziano Pesole. Regularized least squares cancer classifiers from dna microarray data. *BMC Bioinformatics*, 6 (Suppl 4):S2, 2005.

P.J. Green and B.W. Silverman. *Nonparametric Regression and Generalized Linear Models, A Roughness Penalty Approach*. Chapman and Hall, London, 1994.

Marketta Hiissa, Tapio Pahikkala, Hanna Suominen, Tuija Lehtikunnas, Barbro Back, Eija Helena Karsten, Sanna Salanterä, and Tapio Salakoski. Towards automated classification of intensive care nursing narratives. In *The 20th International Congress of the European Federation for Medical Informatics (MIE 2006)*, Maastricht, Netherlands, 2006. To appear.

Maurice G. Kendall. *Rank Correlation Methods*. Griffin, London, 4. edition, 1970.

Carl D. Meyer. *Matrix analysis and applied linear algebra*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.

Tapio Pahikkala, Jorma Boberg, and Tapio Salakoski. Incorporating external information in bayesian classifiers via linear feature transformations. In Tapio Salakoski, Filip Ginter, Sampo Pyysalo, and Tapio Pahikkala, editors, *Proceedings of the 5th International Conference on NLP (FinTAL*

2006), volume 4139 of *Lecture Notes in Computer Science*, pages 399–410, Heidelberg, Germany, 2006a. Springer-Verlag.

Tapio Pahikkala, Filip Ginter, Jorma Boberg, Jouni Järvinen, and Tapio Salakoski. Contextual weighting for support vector machines in literature mining: an application to gene versus protein name disambiguation. *BMC Bioinformatics*, 6(1):157, 2005a.

Tapio Pahikkala, Sampo Pyysalo, Jorma Boberg, Jouni Järvinen, and Tapio Salakoski. Matrix representations, linear transformations, and kernels for natural language processing, 2006b. Submitted.

Tapio Pahikkala, Sampo Pyysalo, Jorma Boberg, Aleksandr Mylläri, and Tapio Salakoski. Improving the performance of bayesian and support vector classifiers in word sense disambiguation using positional information. In Timo Honkela, Ville Könönen, Matti Pöllä, and Olli Simula, editors, *Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning*, pages 90–97, Espoo, Finland, 2005b. Otamedia OY.

Tapio Pahikkala, Sampo Pyysalo, Filip Ginter, Jorma Boberg, Jouni Järvinen, and Tapio Salakoski. Kernels incorporating word positional information in natural language disambiguation tasks. In Ingrid Russell and Zdravko Markov, editors, *Proceedings of the Eighteenth International Florida Artificial Intelligence Research Society Conference*, pages 442–447, Menlo Park, Ca, 2005c. AAAI Press.

Tapio Pahikkala, Evgeni Tsivtsivadze, Jorma Boberg, and Tapio Salakoski. Graph kernels versus graph representations: a case study in parse ranking. In *ECML/PKDD'06 workshop on Mining and Learning with Graphs (MLG'06)*, 2006c. To appear.

Tomaso Poggio and Steve Smale. The mathematics of learning: Dealing with data. *Notices of the American Mathematical Society (AMS)*, 50(5):537–544, 2003.

Marius Popescu. Regularized least-squares classification for word sense disambiguation. In Rada Mihalcea and Phil Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 209–212, Barcelona, Spain, July 2004. Association for Computational Linguistics.

Sampo Pyysalo, Filip Ginter, Juho Heimonen, Jari Björne, Jorma Boberg, Jouni Järvinen, and Tapio Salakoski. Bioinfer: A corpus for information extraction in the biomedical domain, 2006. Submitted.

Ryan Rifkin. *Everything Old Is New Again: A Fresh Look at Historical Approaches in Machine Learning.* PhD thesis, MIT, 2002.

Ryan Rifkin and Aldebaro Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, 2004.

Bernhard Schölkopf, Ralf Herbrich, and Alex J. Smola. A generalized representer theorem. In D. Helmbold and R. Williamson, editors, *Proceedings of the 14th Annual Conference on Computational Learning Theory and and 5th European Conference on Computational Learning Theory*, pages 416–426, Berlin, Germany, 2001. Springer-Verlag. ISBN 3-540-42343-5.

Bernhard Schölkopf and Alexander J. Smola. *Learning with kernels*. MIT Press, Cambridge, MA, 2002.

John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, 2004.

Daniel D. Sleator and Davy Temperley. Parsing english with a link grammar. Technical Report CMU-CS-91-196, Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA, October 1991.

Hanna Suominen, Tapio Pahikkala, Marketta Hiissa, Tuija Lehtikunnas, Barbro Back, Eija Helena Karsten, Sanna Salanterä, and Tapio Salakoski. Relevance ranking of intensive care nursing narratives. In *proceedings of KES2006 10th International Conference on Knowledge-Based & Intelligent Information & Engineering Systems*, 2006. to appear.

J. A. K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Process. Lett.*, 9(3):293–300, 1999.

Evgeni Tsivtsivadze, Tapio Pahikkala, Jorma Boberg, and Tapio Salakoski. Locality-convolution kernel and its application to dependency parse ranking. In Moonis Ali and Richard Dapoigny, editors, *Proceedings of the 19th International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems (IEA/AIE 2006)*, volume 4031 of *Lecture Notes in Computer Science*, pages 610–618, Heidelberg, Germany, 2006. Springer-Verlag.

Evgeni Tsivtsivadze, Tapio Pahikkala, Sampo Pyysalo, Jorma Boberg, Aleksandr Mylläri, and Tapio Salakoski. Regularized least-squares for parse ranking. In A. Fazel Famili, Joost N. Kok, José Manuel Peña, Arno Siebes, and A. J. Feelders, editors, *Proceedings of the 6th International Symposium on Intelligent Data Analysis*, volume 3646 of *Lecture Notes in Computer Science*, pages 464–474, Heidelberg, Germany, September 2005. Springer-Verlag.

V. Vapnik. *Estimation of Dependences Based on Empirical Data [in Russian]*. Nauka, Moscow, 1979. (English translation: Springer Verlag, New York, 1982).

Grace Wahba. *Spline Models for Observational Data*. Series in Applied Mathematics, Vol. 59, SIAM, Philadelphia, 1990.

First order axiomatization of typed feature structures

Richard Elling Moe[⋆]

[⋆]Department of information science and media studies
University of Bergen
Richard.Moe@infomedia.uib.no

Abstract

We axiomatize, in first order logic, the typed feature structures used in constraint based grammar formalism for natural language processing. This is based on a discussion of how type-hierarchies should be interpreted in terms of representation theory. We close with some observations about decidability of feature constraint satisfaction in parsing.

1 Introduction

The history of feature logics can be traced back to terminological logics which originally was meant for knowledge representation in AI but branched into different varieties such as feature logics and, more recently, description logics (Baader et al, 2003) which is used for semantic web ontologies (Baader et al, 2005).

Feature structures first appeared in the 80s for applications in computational linguistics. Specifically in constraint based grammar formalisms (Shieber, 1992) for natural language processing, where they proved useful when side-conditions were added to context-free grammar rules. In the grammar formalism HPSG (Pollard and Sag, 1994) feature structures were combined with types and inheritance hierarchies giving rise to *typed* feature structures (Carpenter, 1992). This approach is now well-established (NLE, 2000) and appears in more recent linguistic frameworks such as the LKB System (Copestake, 2002).

Johnson (1990, 1991) saw feature logics as a fragment of first order logic. In axiomatizing the untyped feature structures he provided a solid mathematical foundation which gave access to the rich body of theory connected to first order logic. He used this to study aspects of decidability. We aim to do the same for the typed variety of feature structures. Finally, in section 8 we briefly consider feature constraints that emerge while parsing. These correspond to a logical expression placing constraints on a single feature structure constructed during the process. The check of whether these constraints can be fulfilled by a feature structure may be reduced to a question of logical satisfiability of the axioms in conjunction with the constraints.

1.1 Preliminaries

Logical formulas are interpreted with relation to *structures*. If L is a first order language then an L-structure is a pair $\langle D, [\![]\!] \rangle$ where D, the domain, is a nonempty set and $[\![]\!]$ interprets the symbols of L by mapping each n-ary function symbol to a total function from D^n to D and each n-ary relation symbol to a relation on D^n, i.e a subset of D^n. The interpretation of a symbol s is denoted by $[\![s]\!]$. The logical symbols have the standard fixed interpretations, where $\equiv$ is the identity relation on D.

We write $\langle D, [\![]\!] \rangle \models_v \varphi$ to express that the formula φ is interpreted as true in $\langle D, [\![]\!] \rangle$ when the free variables are assigned the values specified by a variable assignment function v. In that case φ is said to be *satisfiable*. φ is *valid in* $\langle D, [\![]\!] \rangle$ iff $\langle D, [\![]\!] \rangle \models_v \varphi$ holds for all v. We write $\langle D, [\![]\!] \rangle \models \varphi$ to express this. We write $\langle D, [\![]\!] \rangle \models \Gamma$ to express that each formula in a set Γ is valid in $\langle D, [\![]\!] \rangle$. We say that $\langle D, [\![]\!] \rangle$ is a *model* for Γ.

A formula belongs to the Schönfinkel-Bernay class if it is equivalent to a prenex-formula without function symbols where all existential quantifiers precede the universal quantifiers. That is, formulas of the form $\exists x_1 \ldots \exists x_n \forall y_1 \ldots \forall y_m \varphi$ where φ is a formula with no occurrences of function symbols or quantifiers. We refer to this class as SB. We know a few things about SB-formulas (Dreben and Goldfarb, 1979): Provided they have a model at all, they have a finite model. Satisfiability and validity is decidable and the class of SB-formulas is closed under conjunction.

An *axiomatization* of some phenomenon consists of a set of closed formulas such that its models are precisely the models that bear the characteristics of the phenomenon in question. Thus, an axiomatization describes a class of models and a formula is consistent with the axiomatization iff it has a model in this class.

A *bounded complete partial order* (BCPO) is a partial order $\langle A, \sqsubseteq \rangle$ in which every subset of A with upper bounds has a *least* upper bound. We refer to such sets as being *consistent*. In particular, the empty set is regarded as consistent, having a bottom element $\bot$ as its least upper bound. The functions $\sqcap$ (meet) and $\sqcup$ (join) denote greatest lower bounds and least upper bounds, respectively. Note that $\sqcup$ is partial. Sometimes BCPOs are referred to as *meet semi-lattices*.

2 Feature structures

We adapt Carpenter's definitions (Carpenter, 1992; Copestake, 2000, 2002) concerning typed feature structures, leaving some of his requirements out but otherwise with only minor differences in notation and terminology.

Definition 2.1 *A* type-hierarchy *is a finite BCPO.*

Suppose that $\langle TYPE, \sqsubseteq \rangle$ is a type-hierarchy and $\sigma, \tau \in TYPE$. If $\sigma \sqsubseteq \tau$ we say that σ *subsumes* τ, or alternatively, that σ is *more general/less specific* than τ.

Type-hierarchies are sometimes defined differently. In (Smolka and Aït-Kaci, 1989) and (Smolka et al, 1989), type-hierarchies are viewed as the least quasi-order obtained from a set of subsort declarations in a given signature. Such an order need not be a BCPO.

Types are incorporated into feature structures.

Definition 2.2 *Let $H = \langle TYPE, \sqsubseteq \rangle$ be a type-hierarchy and let $FEAT$ be a set disjoint with $TYPE$. A* feature structure *over H and $FEAT$ is a tuple $\langle Q, \bar{q}, \theta, \delta \rangle$ where:*

- *Q is a set.*
- *$\bar{q} \in Q$ is the* entry
- *$\theta : Q \to TYPE$ is a total* typing *function*
- *$\delta : FEAT \times Q \to Q$ is a partial* transition *function*

A feature structure $\langle Q, \bar{q}, \theta, \delta \rangle$ can be viewed as a directed graph where the elements of Q are nodes, each labelled with the types specified by θ, and where there is an arc, labelled f, from node q to node q' if $\delta(f, q) = q'$. This motivates using graph-theory terms such as 'node', 'path' and 'root' as well as depicting feature structures as graphs.

Carpenter's definition of feature structures is more restrictive than ours in two respects. First, he allows only a finite number of nodes, and secondly, the entry node is required to be a root.

A well-known limitation of first order logic is the impossibility to axiomatize finiteness of a domain, except when explicitly stating the number of elements in it. Hence, we have no hope of axiomatizing feature structures in absolute accordance with Carpenter's definition. However, it is not so obvious that feature structures should be finite. Others, for instance Johnson (1990), allow infinitely many nodes. Carpenter too includes a chapter on the generalisation to infinite feature structures.

Rootedness is defined as the existence of a node from which every other node is reachable. In our case the reachability-relation is the reflexive and transitive closure of the relation $\{\langle d, d' \rangle \mid \delta(f, d) = d' \text{ for some } f \in FEAT\}$. Another limitation of first order logic is that reflexive and transitive closures can not be axiomatized. Nor can we then axiomatize rootedness. At best, we could specify a relation that contains the closure. This approach is taken in the description logic SHIQ. However, it is intended for use as an ontology language for the semantic web (Baader et al, 2005). A main concern is then to strike a balance between expressiveness and the computational complexity of reasoning mechanisms and so it is acceptable to settle for 'half-way' solutions. This option is not open to us since our purpose is to provide a precise account rather than just trying to make the best of things. Again we escape the difficulties by observing that not all accounts agree on the question of rootedness. For instance, Johnson (1990) allows disconnected feature structures, while the PATR grammar formalism asserts the existence of a root (Shieber, 1992).

Thus, the framework of first order logic forces us to choose the more general approach, allowing infinite feature structures and dropping the requirement that the designated entry node should be a root.

An important use of types in feature structures lies in specifying that only certain nodes may possess a given feature.

Definition 2.3 *An* appropriateness specification *over the type-hierarchy $\langle TYPE, \sqsubseteq \rangle$ and features $FEAT$ is a partial function $\mathcal{A} : FEAT \times TYPE \to TYPE$ that meets the following conditions:*

- *For every feature $f \in FEAT$, there is a most general type $Intro(f) \in TYPE$ such that $\mathcal{A}(f, Intro(f))$ is defined.*
- upward closure and right monotonicity*: If*

$\mathcal{A}(f, \sigma)$ is defined and $\sigma \sqsubseteq \tau$, then $\mathcal{A}(f, \tau)$ is also defined and $\mathcal{A}(f, \sigma) \sqsubseteq \mathcal{A}(f, \tau)$.

We may refer to $\langle TYPE, \sqsubseteq\rangle$ and $\mathcal{A}$ as a typing scheme *(over $FEAT$).*

Appropriateness-specification form a basis for coherence-conditions, such as the following (Carpenter, 1992).

Definition 2.4 *Let $\langle TYPE, \sqsubseteq\rangle$ be a type-hierarchy and let $\mathcal{A}$ be an appropriateness specification. A feature structure $\langle Q, \bar{q}, \theta, \delta\rangle$ is said to be* well-typed *if whenever $\delta(f, q)$ is defined, $\mathcal{A}(f, \theta(q))$ is defined and $\mathcal{A}(f, \theta(q)) \sqsubseteq \theta(\delta(f, q))$.*

Well-typedness is sometimes accompanied by the requirement that all appropriate transitions should be present.

Definition 2.5 *Let $\langle TYPE, \sqsubseteq\rangle$ be a type-hierarchy and let $\mathcal{A}$ be an appropriateness specification over $FEAT$. A feature structure $\langle Q, \bar{q}, \theta, \delta\rangle$ is* totally well-typed *if and only if it is well-typed and if $q \in Q$ and $f \in FEAT$ are such that $\mathcal{A}(f, \theta(q))$ is defined, then $\delta(f, q)$ is defined.*

3 Typed domains

In this section we address the question of how to embed type-hierarchies within the framework of first order logic. We want to view feature structures as certain first order structures with domains consisting of nodes. Then we must somehow impose the structure of the hierarchy onto the domain. We find the basis for our approach in *representation theory* (Davey and Priestley, 1991) where types are interpreted as sets in a natural way. This view has appeared in connection with a variety of ordering-relations, for instance in (Davey and Priestley, 1991; Smolka and Aït-Kaci, 1989; Smolka et al, 1989; Smolka, 1992). Carpenter (1992), however, does not explicate his type-hierarchies in these terms. So, we must address this question and need to resolve two issues: Should inconsistent types necessarily be associated with disjoint sets? Smolka (1992) requires that a top element $\top$ should be interpreted as the empty set and that inconsistent types are interpreted as disjoint sets. On the other hand (Smolka and Aït-Kaci, 1989) and (Smolka et al, 1989) allow intersecting interpretations of inconsistent types. We shall discuss these alternatives. Initially we adopt the more general approach allowing intersecting interpretations of inconsistent types and nonempty interpretation of $\top$, if present, before considering tightening the interpretation to consistent typing in the same vein as Smolka.

The second issue to address is whether the first order representation of types should be isomorphic to the original hierarchy. Let us begin by developing a straightforward representation theory for the type-hierarchies. At the core lies the basic definition of an interpretation of a type-hierarchy.

Definition 3.1 *If $H = \langle TYPE, \sqsubseteq\rangle$ is a type-hierarchy and D is a set then a D-*interpretation of *H is a function $f : TYPE \to 2^D$ assigning a subset of D to every type. Let D^f_σ denote the set assigned to a type σ by a D-interpretation f.*

Sometimes it is convenient to illustrate interpretations of simple type-hierarchies by projecting the Euler-diagram of the interpreting sets onto the Hasse diagram representing the type-hierarchy, as shown in figure 1. However, we occasionally find it necessary to add a few comments to make such an illustration clear.

Definition 3.1 does not take into account the inheritance-information represented in the type-hierarchy. To obtain close correspondence between the type-hierarchy and its interpretation we need additional requirements. Given a D-interpretation f of a type-hierarchy $H = \langle TYPE, \sqsubseteq\rangle$ we regard the partially ordered set $\langle\{D^f_\sigma \mid \sigma \in TYPE\}, \supseteq\rangle$ as a *representation* of H. We shall look closer at two representation schemes with different degrees of correspondence with the type-hierarchy. Such representations will be referred to as *typed domains*. A typed domain is said to be *faithful* if it is isomorphic to the original hierarchy.

3.1 Faithful representation

We shall interpret type-hierarchies as sets such that the subset-relation respects the partial order in the type-hierarchy. We do this by requiring that joins should correspond precisely to intersections of type-interpretations. Moreover, we want the interpretation of the bottom element to correspond to the universal type. That is, it should comprise 'everything' by demanding that a D-interpretation associates $\bot$ with the entire set D.

Definition 3.2 *If D is a nonempty set, $H = \langle TYPE, \sqsubseteq\rangle$ is a type-hierarchy and f a D-interpretation of H then f defines a* faithfully H-typed domain *iff the following conditions are satisfied:*

- $D^f_\bot = D$

- $\sqcup\Sigma = \tau$ *iff* $\bigcap_{\sigma\in\Sigma} D^f_\sigma = D^f_\tau$, *where* $\{\tau\} \cup \Sigma \subseteq TYPE$.

Note that $\sigma \sqsubseteq \tau$ iff $\sqcup\{\sigma,\tau\} = \tau$, so as a special case of definition 3.2 we get $\sigma \sqsubseteq \tau$ iff $D^f_\tau \subseteq D^f_\sigma$.

Figure 1 illustrates an interpretation which satisfies definition 3.2.

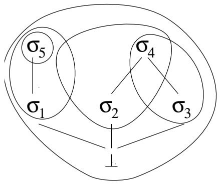


Figure 1: Interpretation satisfying definition 3.2. Note that the interpretation of σ_4 is the intersection between the interpretations of σ_2 and σ_3.

Inconsistency of types appears as a non-existing join in a type-hierarchy H, whereas in a faithfully H-typed domain defined by f, this inconsistency is reflected by their intersection not being assigned to a type by f. Note also that in order to satisfy definition 3.2, f can assign the empty set to no type other than a top element of H.

The representation defined by an interpretation satisfying definition 3.2 is in fact a consistently complete partial order. Specifically, $\langle\{D^f_\sigma \mid \sigma \in TYPE\}, \supseteq\rangle$ is a (finite) BCPO, with $\bigcap$ as join-function. The meet of a set $X \subseteq \{D^f_\sigma \mid \sigma \in TYPE\}$ corresponds to

$$min_{\subseteq}\{Y \mid Y \in \{D^f_\sigma \mid \sigma \in TYPE\}, Y \supseteq \bigcup_{x\in X} x\}$$

Having established the particulars of the BCPO induced on a domain by an interpretation satisfying definition 3.2, we verify that it retains the original hierarchy.

Theorem 3.1 *If* $H = \langle TYPE, \sqsubseteq\rangle$ *is a type-hierarchy,* f *a* D*-interpretation of* H *and* f *defines a faithfully* H*-typed domain then* $\langle\{D^f_\sigma \mid \sigma \in TYPE\}, \supseteq\rangle$ *and* $\langle TYPE, \sqsubseteq\rangle$ *are isomorphic.*

The type-inheritance described by a type-hierarchy H and a faithfully H-typed domain correspond exactly to each other since they are isomorphic structures. Furthermore, the corresponding meet and join functions imitate each other in every detail. Thus, as far as subsumption, meet and join are concerned, faithfully typed domain provides perfect representation of type-hierarchies.

3.2 Degenerated representation

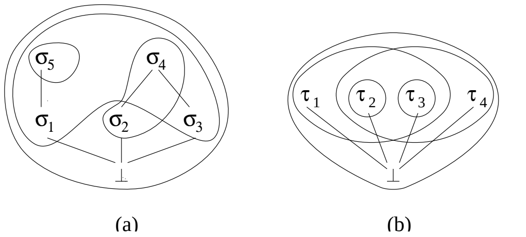
In this section we relax the requirement that typed domains should be faithful. There can be no isomorphic representation unless there are enough elements available to form different interpretations for all the types. Even if a representation contains a sufficient number of elements, the type-interpretations may be organised in a degenerated way.

In our context, type-representations shall be populated with nodes of feature structures. It may well be that a feature structure has too few nodes to obtain a faithful representation, in which case we may have to settle for a *degenerated* one. We use the term 'typed domain', dropping the qualifier 'faithful', when referring to such repesentation.

Definition 3.3 *If* $H = \langle TYPE, \sqsubseteq\rangle$ *is a type-hierarchy and* f *a* D*-interpretation of* H *then* f *defines an* H*-*typed domain *iff the following conditions are satisfied:*

- $D^f_\perp = D$
- $\sqcup\Sigma = \tau$ *implies* $\bigcap_{\sigma\in\Sigma} D^f_\sigma = D^f_\tau$, *where* $\{\tau\} \cup \Sigma \subseteq TYPE$.

As a special case of definition 3.3 we get that $\sigma \sqsubseteq \tau$ implies $D^f_\tau \subseteq D^f_\sigma$. Figures 2 and 3 illustrate interpretations which satisfy definition 3.3 (but not definition 3.2).



(a)

(b)

Figure 2: interpretations satisfying definition 3.3. Note that the interpretation of σ_4 is the intersection between the interpretations of σ_2 and σ_3.

On this approach, the ordering-relation is preserved but distinct types may be indistinguishable. Information about consistency may be lost and the representation need not even be a BCPO.

In figure 2(a) the types σ_1 and σ_3 are interpreted as the same set. Neither does definition 3.3 prohibit that the interpretation of a type σ contains the interpretation of a type incomparable with σ. Because of this, a typed domain may hold less information about consistency than the type-hierarchy it represents. Furthermore, definition 3.3 does not guarantee that the corresponding representation is a BCPO. For instance, let f be the D-interpretation that defines the typed domain of figure 2(b). In the representation, both $D^f_{\tau_2}$ and $D^f_{\tau_3}$ are upper bounds for $\{D^f_{\tau_1}, D^f_{\tau_4}\}$, but there is no *least* upper bound for this set, so the BCPO-requirements are not met. Under these circumstances we can not expect a representation to be isomorphic to the type-hierarchy. On the other hand, the ordering relation will be preserved.

Lemma 3.1 *If $H = \langle TYPE, \sqsubseteq\rangle$ is a type-hierarchy and f a D-interpretation that defines an H-typed domain then f is a monotone map from $\langle TYPE, \sqsubseteq\rangle$ to $\langle\{D^f_\sigma \mid \sigma \in TYPE\}, \supseteq\rangle$*

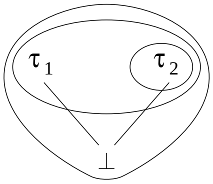


Figure 3: interpretation satisfying definition 3.3

The following illustrates that least upper bounds and greatest lower bounds are not preserved. Assume that f is the D-interpretation that defines the typed domain of figure 3, then

- The greatest lower bound of $\{\tau_1, \tau_2\}$ is $\perp$ but the greatest lower bound of $\{D^f_{\tau_1}, D^f_{\tau_2}\}$ is $D^f_{\tau_1}$
- The least upper bound of $\{\tau_1, \tau_2\}$ is not defined while the least upper bound of $\{D^f_{\tau_1}, D^f_{\tau_2}\}$ is $D^f_{\tau_2}$

Having to live with the fact that a feature structure may contain too few nodes to provide a faithful representation, we may find comfort in that it is always possible to turn a degenerated representation into an isomorphic one while preserving the elements' memberships in type-interpretations. The transformation consists in simply adding 'junk'-elements to support the missing types.

In order to be able to talk about parts of a typed domain or the addition of elements, we introduce the concept of expansion/subinterpretation.

Definition 3.4 *Let $H = \langle TYPE, \sqsubseteq\rangle$ be a type-hierarchy and let f be a D-interpretation of H. An* expansion *of f is a D'-interpretation f' such that*

- $D \subseteq D'$
- *$d \in D^f_\sigma$ iff $d \in {D'}^{f'}_\sigma$ for all $d \in D$ and $\sigma \in TYPE$.*

We say that f defines a subinterpretation of f'.

If an expansion defines a faithfully typed domain it is said to be a *faithful expansion*.

Theorem 3.2 *Let H be a type-hierarchy and let f be a D-interpretation of H. f defines an H-typed domain iff there is f' defining a faithfully H-typed domain such that f is a subinterpretation of f'*

4 Typing functions

Recall our definition (2.2) of feature structures, where every node is associated with a type by means of a typing function. In this section we develop the corresponding idea for typed domains.

Obviously, an appropriate typing function should map a node to a type whose interpretation contains the node, but as we have seen in the previous sections, an element x of a typed domain may belong to the interpretation of several types. Some of these types may even be incomparable and therefore not well suited to be the designated type associated with x, unless some kind of overriding is employed. However, there is always a greatest type that offers a compromise solution to this difficulty. Specifically, the greatest type comparable with every other type whose interpretation contains x. Hence, for any element x, we can identify a unique designated type, referred to as *the* type of x. This provides a typing function.

Definition 4.1 *Let $H = \langle TYPE, \sqsubseteq\rangle$ be a type-hierarchy and let f be a D-interpretation that defines an H-typed domain. The* typing function *θ is defined by*

$$\theta(x) = max_{\sqsubseteq}\{\sigma \mid x \in D^f_\sigma, \text{ for every type } \tau\text{:} \\ x \in D^f_\tau \text{ implies } \sigma \sqsubseteq \tau \text{ or } \tau \sqsubseteq \sigma\}$$

The typing function is illustrated in figure 4, where the interpretation of σ_5 is the intersection between the interpretations of σ_2 and σ_3. Note that d_2 is contained in the interpretations of both σ_4 and σ_5, neither of which is more suited than the other to be its designated type. As a 'compromise', θ assigns d_2 to σ_1.

Nodes that are not shared by interpretations of inconsistent types are easier to deal with. Then the type of the node is simply the join of the types whose interpretations the node belongs to. Moreover, this is the greatest such type.

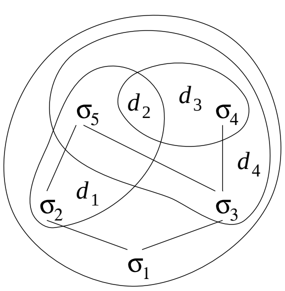


Figure 4: $\theta(d_1) = \sigma_2$, $\theta(d_2) = \sigma_1$, $\theta(d_3) = \sigma_4$, $\theta(d_4) = \sigma_3$

Lemma 4.1 *Let $H = \langle TYPE, \sqsubseteq \rangle$ be a type-hierarchy, let f be a D-interpretation defining an H-typed domain and let θ be the corresponding typing function according to definition 4.1. Let $d \in D$ and let $\Sigma_d = \{\sigma \mid d \in D^f_\sigma\}$. If Σ_d is consistent then $\sqcup\Sigma_d = max_\sqsubseteq \Sigma_d$ and $\theta(d) = max_\sqsubseteq \Sigma_d$*

We will get back to this observation in section 6.

5 First order typed feature structures

In this section we define typed feature structures in the framework of first order logic. The idea is to identify feature structures among the first order structures described in section 1.1. Domains will consist of nodes and we let the domains be typed in order to incorporate the type-hierarchy.

In section 3.2 we noted that it takes a certain number of elements to obtain an isomorphic representation of the type-hierarchy, and that a particular feature structure may have too few nodes. So, we can not expect the domain of a first order feature structure to be faithfully typed. This fact does not reflect any conflict between the feature structure and the type-hierarchy it is supposed to obey, only that there may be too few types involved to make use of the whole hierarchy. However, applying theorem 3.2 we find that any first order typed feature structure can be expanded to, and preserved within, a faithfully typed one.

Theorem 5.1 *Let $H = \langle TYPE, \sqsubseteq \rangle$ be a type-hierarchy and let $\langle D, [\![]\!] \rangle$ be a structure over $\widehat{TYPE} \cup \widehat{FEAT} \cup \{entry\}$. $\langle D, [\![]\!] \rangle$ is an H-typed feature structure iff it is a substructure of a faithfully H-typed feature structure $\langle D', [\![]\!]' \rangle$*

In view of this we do not demand that the typed domain incorporated in the structure should be faithful.

First order structures are defined with respect to a language and, in our case, the language must contain symbols representing the types and features. Types will be treated as properties of nodes so our language must contain a unary relation-symbol $\hat{\sigma}$ for every $\sigma \in TYPE$. Clearly, a feature can be seen as a partial function on the set of nodes. However, the function symbols of classical first order logic are required to be interpreted as *total* functions. We conform to this tradition by letting binary relation symbols, rather than unary function symbols, represent the features, at the expense of having to state their functional nature explicitly. Thus, for each feature f we include a binary function-symbol $\hat{f}$. We let $\widehat{TYPE}$ and $\widehat{FEAT}$ denote the sets of type- and feature-symbols. Finally we include in our language a constant $entry$ to represent the entry node.

A first order structure $\langle D, [\![]\!] \rangle$ interprets all non-logical symbols by means of $[\![]\!]$. Sometimes we wish to focus on the interpretation of types only. For this purpose we let $[\![]\!]_T$ denote the restriction of $[\![]\!]$ to $\widehat{TYPE}$.

Now we are ready to give a definition of typed feature structures in the framework of first order logic.

Definition 5.1 *Let $H = \langle TYPE, \sqsubseteq \rangle$ be a type-hierarchy, $FEAT$ a set of feature and $\langle D, [\![]\!] \rangle$ a first order structure over $\widehat{TYPE} \cup \widehat{FEAT} \cup \{entry\}$. Then $\langle D, [\![]\!] \rangle$ is a* first order typed feature structure (over H and $FEAT$) *iff the following conditions are satisfied*

- *$[\![]\!]_T$ defines an H-typed domain.*
- *For all $f \in FEAT$ and all $d \in D$ there is at most one $d' \in D$ such that $\langle d, d' \rangle \in [\![\hat{f}]\!]$*

In order to specify when a first order typed feature structure is well-typed etc, we need to take a closer look at the connection between our two definitions of feature structures and see how we can view the new definition in light of the old one (2.2). For this purpose we translate first order typed feature structures into ordinary ones, i.e. according to definition 2.2.

Definition 5.2 *Consider H, $FEAT$ and $entry$ of definition 5.1 and let $S = \langle D, [\![]\!] \rangle$ be a first order structure over $\widehat{TYPE} \cup \widehat{FEAT} \cup \{entry\}$. The* translation of *$S$ is the tuple $\langle D, [\![entry]\!], \theta, \delta \rangle$ where $\theta : D \to TYPE$ and $\delta : FEAT \times D \to D$ are defined by*

$$\theta(x) = max_\sqsubseteq\{\sigma \mid x \in [\![\hat{\sigma}]\!]_T, \textit{ for every } \tau\textit{:} \\ x \in [\![\hat{\tau}]\!]_T \textit{ implies } \sigma \sqsubseteq \tau \textit{ or } \tau \sqsubseteq \sigma\}$$

and

$$\delta(f, d) = \begin{cases} d' & \textit{if } \langle d, d'\rangle \in [\![\hat{f}]\!] \\ \textit{undefined} & \textit{otherwise} \end{cases}$$

The translation of a first order typed feature structure is a proper feature structure according to definition 2.2, but the converse does not hold. When applied to a structure that does not satisfy definition 5.1, the translation may well produce a feature structure.

The definitions of properties concerning our original feature structures can be transferred to the first order typed feature structures by means of the translation. I.e. a first order typed feature structure is well-typed iff its translation is well-typed, and similarly for other properties of feature structures.

5.1 A mismatch

We have just seen that every first order typed feature structure can be translated into an ordinary typed feature structure, so we may view a first order typed feature structure as a representation of an ordinary one. It should also be reasonably clear that every ordinary feature structure has such a representation. However, there is an unfortunate circumstance in that the translation is not one to one. An ordinary feature structure may be represented by several first order feature structures. Take for instance the feature structure $\langle\{q_0, q_1, q_2\}, q_0, \theta, \delta\rangle$, depicted in figure 5, over a type-hierarchy H and features $\{f_1, f_2\}$.

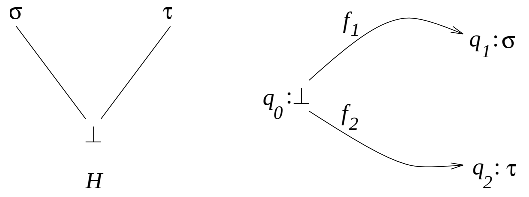


Figure 5: a feature structure over H with entry q_0

Now, let $\langle\{q_0, q_1, q_2\}, [\![]\!]\rangle$ and $\langle\{q_0, q_1, q_2\}, [\![]\!]'\rangle$ be first order H-typed feature structures such that $[\![entry]\!] = [\![entry]\!]' = q_0$, $[\![f_1]\!] = [\![f_1]\!]' = \{\langle q_0, q_1\rangle\}$ and $[\![f_2]\!] = [\![f_2]\!]' = \{\langle q_0, q_2\rangle\}$. Also, let $[\![]\!]_T$ define the H-typed domain depicted in figure 6(a) and let $[\![]\!]'_T$ define the H-typed domain depicted in figure 6(b). In both cases the corresponding typing function maps q_0 to $\bot$, q_1 to σ and q_2 to τ. Hence we have two different first order typed feature structures with the same translation. The cause of this effect will always be a nonempty intersection between interpretations of inconsistent types.

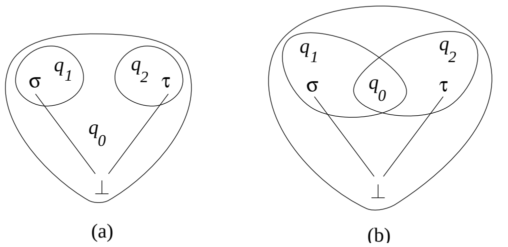


Figure 6: $\{q_0, q_1, q_2\}$-interpretations of H with equal typing functions

Lemma 5.1 *Let $H = \langle TYPE, \sqsubseteq\rangle$ be a type-hierarchy and $FEAT$ a set of features. Let $\langle D, [\![]\!]\rangle$ and $\langle D, [\![]\!]'\rangle$ be distinct first order typed feature structures over $\widehat{TYPE} \cup \widehat{FEAT} \cup \{entry\}$. If they have the same translation then there is $d \in D$ such that either $\{\sigma \mid d \in [\![\hat{\sigma}]\!]\}$ or $\{\sigma \mid d \in [\![\hat{\sigma}]\!]'\}$ is inconsistent.*

5.2 Well-typedness under inconsistency

First order typed feature structures are defined in terms of the ordinary feature structures such that properties are translated into the first order framework. However, this does not always work as well as we would like.

It turns out that in translation well-typedness becomes a stronger requirement than intended. The reason is that the typing function may have to pick a more general type than desired when a node belongs to the interpretations of inconsistent types. Consider the type-hierarchy in figure 4. Since the d_2 is contained in the interpretations of both σ_4 and σ_5 , the typing function will associate it with the compromise type σ_1. But, there may be features specific for σ_4 and σ_5, which are not appropriate for σ_1. This is unfortunate in the context of well-typedness, where a feature is required to be appropriate for *the* type of the node to which it belongs. Clearly, well-typedness of a first order typed feature structure may require that reasonable features for a node are omitted. There is a similar problem for the values of features.

These problems are rooted in the inconsistent typing scheme we adopted initially. Compromise types can not occur under consistent typing since then the type of a node corresponds to a unique maximal type. Cf lemma 4.1.

6 Consistently typed domains

We now consider tightening our typing scheme by demanding that the interpretations of inconsistent types should be disjoint. When we place these requirements on our typed domains we refer to them as being *consistently* typed.

Definition 6.1 *Let $H = \langle TYPE, \sqsubseteq\rangle$ be a type-hierarchy and let f be a D-interpretation that defines an H-typed domain. f is said to define a* consistently *H-typed domain iff for every inconsistent $\Sigma \subseteq TYPE$, $\bigcap_{\sigma\in\Sigma} D^f_\sigma = \emptyset$*

We have seen several indications that consistent typing has advantages in our setting. First, the typing function is simplified. It follows from lemma 4.1, that if f is a D-interpretation defining a consistently typed domain then $\theta(d) = max_{\sqsubseteq}\{\sigma \mid d \in D^f_\sigma\}$ for any $d \in D$.

This in turn removes the problems, reported in section 5.2, of well-typedness being a stronger condition than intended.

The HPSG grammar formalism is governed by the so-called *modelling convention* which involves the following, as Carpenter (1992) puts it: 'According to this convention, the nodes of a feature structure are taken to represent objects, and we assume that every node is labelled with a type symbol which represents the most specific conceptual class to which the object is known to belong.' This convention obviously corresponds precisely to θ under consistent typing.

In section 5.1 we saw that the translation of first order feature structures is not one to one. When we restrict ourselves to consistently typed feature structures we obtain this property. That is, an ordinary typed feature structure is the translation of at most one first order typed feature structure.

Lemma 6.1 *Let $H = \langle TYPE, \sqsubseteq\rangle$ be a type-hierarchy. Let $\langle D, \llbracket\rrbracket\rangle$ and $\langle D', \llbracket\rrbracket'\rangle$ be first order typed feature structures such that $\llbracket\rrbracket_T$ and $\llbracket\rrbracket'_T$ defines consistently H-typed domains and let F and F' be their translations, respectively. If $F = F'$ then $\langle D, \llbracket\rrbracket\rangle = \langle D', \llbracket\rrbracket'\rangle$.*

The translation is not only one to one, it is also onto. The construction of a consistently typed first order feature structure which translates to a given ordinary feature structure is a straightforward matter. Hence, translation defines a one to one correspondence.

Apparently, consistent typing is the solution to all the problems reported above. We conclude that first order typed feature structures should have consistently typed domains and confine the remaining discussion accordingly.

7 Axiomatization

Having defined the first order typed feature structures and settled for a degenerated and consistent representation of the type-hiearchy, their axiomatization is relatively straightforward.

In the follwing, let $H = \langle TYPE, \sqsubseteq\rangle$ be a type-hierarchy, $\mathcal{A}$ an appropriateness specification over H and $FEAT$, and let L be a first order language over $\widehat{TYPE} \cup \widehat{FEAT} \cup \{entry\}$. Our goal is to develop axiomatizations in the form of sets of L-formulas.

Starting with the typed domains we define Γ_H to be the set

$$\{\forall x \hat{\bot}(x)\} \cup \{\forall x(\hat{\tau}(x) \leftrightarrow \bigwedge_{\sigma\in\Sigma} \hat{\sigma}(x)) \mid \sqcup\Sigma = \tau\}$$

which constitutes a set of axioms for typed domains

Axioms for consistent typing are now obtained by simply extending Γ_H with axioms that keep the interpretations of inconsistent types disjoint.

$$\Gamma_c = \{\forall x \neg(\bigwedge_{\sigma\in\Sigma} \hat{\sigma}(x)) \mid \text{ inconsistent } \Sigma \subseteq TYPE\}$$

The models for $\Gamma_c \cup \Gamma_H$ are precisely those where the interpretation of type symbols forms a consistently H-typed domain.

We now turn to the features. The interpretation of the relation symbols in $\widehat{FEAT}$ must be restricted to correspond to partial functions. Define Γ_{feat} to be the set

$$\{\forall x \forall y \forall z((\hat{f}(x,y) \wedge \hat{f}(x,z)) \rightarrow y \equiv z) \mid f \in FEAT\}$$

It should be clear that the models for $\Gamma_H \cup \Gamma_c \cup \Gamma_{feat}$ are precisely the first order consistently typed feature structures.

We add axioms to capture well-typedness. Define Γ_W to be the set

$$\begin{array}{l} \Gamma_{feat} \cup \Gamma_H \cup \Gamma_c \cup \\ \{\forall x((\exists y\, \hat{f}(x,y)) \rightarrow \hat{\iota}(x)) \mid \iota = Intro(f)\} \cup \\ \{\forall x \forall y((\hat{f}(x,y) \wedge \hat{\tau}(x)) \rightarrow \hat{\tau}'(y)) \mid \mathcal{A}(f,\tau) = \tau'\} \end{array}$$

Γ_W is indeed an axiomatization of well-typed feature structures.

Theorem 7.1 *Let $\langle D, \llbracket\rrbracket\rangle$ be a structure over $\widehat{TYPE} \cup \widehat{FEAT} \cup \{entry\}$.*

$\langle D, \llbracket\rrbracket\rangle$ is a consistently well-typed feature structure iff $\langle D, \llbracket\rrbracket\rangle \models \Gamma_W$

Finally we extend to *total* well-typedness. Define Γ_{TW} to be the set

$$\Gamma_W \cup \{\forall x(\hat{\sigma}(x) \to \exists y \hat{f}(x,y)) \mid \mathcal{A}(f,\sigma) \text{ defined}\}$$

which provides the axiomatization.

Theorem 7.2 *Let* $\langle D, [\![]\!] \rangle$ *be a structure over* $\widehat{TYPE} \cup \widehat{FEAT} \cup \{entry\}$.

$\langle D, [\![]\!] \rangle$ *is a consistently and totally well-typed feature structure iff* $\langle D, [\![]\!] \rangle \models \Gamma_{TW}$

Provided with axiomatizations we can easily cash in some results about typed feature structures. Let us take a look at their use in natural language parsing.

8 Feature constraints, satisfaction and decidability

When parsing, information about constituents is gathered through accumulation of the feature structures encountered during the process. The forming of a phrase from subconstituents imposes constraints on this accumulation such that the question of whether the phrase is grammatically well formed relies on whether the feature structures may be unified into one in a way that respects the constraints.

First order formulas are also suitable for feature description and in Johnson (1991) the parsing process accumulates information and requirements about phrase-constituents using formulas instead of feature structures. I.e these formulas place constraints on the construction of a single feature structure and we refer to them as feature constraints. Then, unification corresponds roughly to a logical conjunction. A phrase is admitted as grammatically well formed if its accumulated feature constraint is satisfiable by a feature structure. That is, if the feature constraint in conjunction with the axiomatization of feature structures is satisfiable in the sense of first order logic.

From a computational point of view, it is of course important that the question of satisfiability is decidable. However, this is not always the case. Undecidability occurs in grammar-formalisms where functional uncertainty is incorporated in the constraint-language, in the form of regular expressions in path description. Then, if negation is also admitted, the language becomes too expressive, and the question of satisfiability is thus rendered undecidable. Consult Keller (1993) and Baader et al (1993).

In our case, the axioms for well-typed feature structures are completely within the SB-fragment of first order logic, for which satisfiability is known to be decidable. I.e. the satisfiability of $\Gamma_W \cup \{\varphi\}$ is decidable for any formula φ in SB. Hence, the feature constraint language may be SB entirely and still remain decidable. The axioms for totally well-typed feature structures goes beyond SB so we do not get decidability results for free. Actually, the feature constraints or the type-hierarchy must be restricted in order to retain decidability, but that is an other story.

9 Conclusion

Our goal has been to axiomatize Carpenter's typed feature structures within first order logic. The restrictions of the logical framework forced a confinement to a more general variety of feature structures, abandoning requirements for finiteness and rootedness.

We have observed that a particular feature structure may contain too few nodes to support the entire type hierarchy and we therefore relaxed the requirement that first order feature structures should contain an isomorphic representation of it.

We have discussed whether the typing scheme should allow the interpretations of inconsistent types to intersect. If so, the notion of well-typedness is apparently a stronger condition than Carpenter intended it to be. This suggests that consistent typing is preferable. We find another indication when we consider the modelling convention of HPSG in light of lemma 4.1. The labeling of each node with a type symbol which represents the most specific conceptual class to which the node is known to belong, corresponds precisely to the typing function in a consistently typed domain. Finally, consistent typing provides for a one to one correspondence between first order feature structures and the ordinary ones. We conclude that the consistent typing is the appropriate interpretation scheme of type hierarchies in our context.

We have presented axioms for well-typed and totally well-typed feature structures. The checking of constraints in constraint-based parsing can be reduced to checking the satisfiability of these axioms in conjunction with corresponding expressions in a feature constraint language. Since the axioms for well-typed feature structures are within the SB-fragment we have established that the satisfiability-check is decidable. Whereas the axioms for totally well-typed feature structures fall outside SB and we may not be allowed the full expressive power of SB for feature constraints.

References

Franz Baader, Hans-Jürgen Bürckert, Bernhard Nebel, Werner Nutt, Gert Smolka. On the Expressivity of Feature Logics with Negation, Functional Uncertainty, and Sort Equations. *Journal of Logic, Language and information* volume 2, No. 1, 1993

Franz Baader, Diego Calvanese, Deborah McGuiness, Daniele Nardi, Peter Patel-Schneider (eds). *The description logic handbook*. Cambridge University Press, 2003

Franz Baader, Ian Horrocs, Ulrike Sattler. Description logics as ontology languages for the semantic web. In D. Hutter, W. Stephan. Mechanizing mathematical reasoning. LNAI 2605, Springer Verlag 2005

Bob Carpenter. *The Logic of Typed Feature Structures*. Cambridge University Press 1992

Ann Copestake. Definitions of typed feature structures. *Natural Language Engineering* 6 (1), Cambridge University Press 2000

Ann Copestake. *Implementing typed feature structures grammars*. CSLI Publications, 2002

B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press 1991

Burton Dreben, Warren D. Goldfarb. *The Decision Problem: Solvable Classes of Quantificational Formulas.* Addison-Wesley 1979

Mark Johnson. Expressing Disjunctive and Negative Feature Constraints with Classical First-Order Logic. The Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics 1990

Mark Johnson. Features and Formulae. *Computational Linguistics* 17(2) 1991

Bill Keller. *Feature Logics, Infinitary Descriptions and Grammar*. CSLI Lecture Notes Number 44, 1993

NLE Special issue on HPSG Parsing. *Natural Language Engineering* 6 (1), 2000

Carl Pollard, Ivan Sag. *Head-driven phrase structure grammar*. Chicago University Press, 1994

Stuart M. Shieber. *Constraint-Based Grammar Formalisms*. The MIT Press, 1992

Gert Smolka. A Feature Logic with Subsorts. In J. Wedekind and C. Rohrer (eds.) *Unification in Grammar*. The MIT press, 1992

Gert Smolka, Hassan Aït-Kaci. Inheritance Hierarchies: Semantics and Unification. *Journal of Symbolic Computation* 7, 1989

Gert Smolka, Werner Nutt, Joseph A. Goguen,José Meseguer. Order-Sorted Equational Computation. In Aït-Kaci, Nivat. *Resolution of Equations in Algebraic Structures* Volume 2. Academic Press 1989

Emergence of multilingual representations by independent component analysis using parallel corpora

Jaakko J. Väyrynen[⋆]
[⋆]Adaptive Informatics Research Centre
Helsinki University of Technology
P.O. Box 5400, FIN-02015 TKK, FINLAND
Jaakko.J.Vayrynen@TKK.Fi

Tiina Lindh-Knuutila[†]
[†]Adaptive Informatics Research Centre
Helsinki University of Technology
P.O. Box 5400, FIN-02015 TKK, FINLAND
Tiina.Lindh-Knuutila@TKK.Fi

Abstract

This paper reports the first results on extracting a meaningful representation for words from multilingual parallel corpora. Independent component analysis is used to extract a number of components from statistics calculated for words in contexts. Individual components are meaningful and multilingual and words are represented as a bag of concepts model. The component space created by the extracted components is also multilingual. Words that are related in different languages appear close to each other in the component space, which makes it possible to find translations for words between languages.

1 Introduction

Analysis of symbolic text using statistical methods requires extraction of a numerical representation or statistics. Contextual information has been widely used in statistical analysis of natural language corpora (Deerwester et al., 1990; Honkela et al., 1995; Ritter and Kohonen, 1989). One useful representation for words can be constructed by taking into account the contexts in which they occur. This is based on the distributional hypothesis, which states that words that occur in similar contexts tend to be similar. The length of the context may vary from just neighboring words to the sentence and to the whole document, and thus representing different similarities between words. In this paper, we have contexts that span aligned sentences in different languages and word co-occurrences are calculated inside these contexts. Another approach to produce statistics is to count how many times each word occurs in each document.

We follow here the ideas that knowledge of language emerges from language use and knowledge of language is conceptualization. Knowledge of language includes, for instance, grammar, semantics and similarity of words. As an example of language use, we take a text corpus that represents actual language use and calculate statistics that we try to refine into a compact and meaningful representation.

For the statistics, we study frequencies of words in contexts. Independent component analysis (ICA) is used in the extraction of a representation for words from the collected statistics. ICA is an unsupervised method that finds a feature representation for each word, where the values of the features are continuous and several features can be present in one word at the same time. Unsupervised methods have the advantage that they do not need an explicit training set where a correct representation is known. On the other hand, the representations found by unsupervised methods are emergent, i.e., they follow the data instead of rules used by humans. Sahlgren and Karlgren (2005) applied a method called random indexing to find translations of words from a bilingual parallel corpus. The approach there was to use aligned paragraphs as the contexts and to assign a random vector to each context. The relations of word vectors were examined to find translations of words. Our aim is to find also a meaningful representation that encodes linguistic knowledge.

We hope that independent component analysis can find an emerging structure for the data, providing a meaningful representation for the words. We expect the word representations to have related words closer to each other than unrelated words, and that the emergent features to be meaningful. Especially, we want to see translations of words.

The use of multilingual parallel corpora gives a possibility to analyze how meaningful the features found by ICA are. In order to measure the success of our method and to show its potential, we demonstrate its use using a parallel corpus that has the same text in different languages and in which regions of texts are matched together. The initial experiments were carried out using bilingual material.

2 Data and methods

In this section, we describe our data, a sentence-aligned parallel corpus of Finnish-English text, collection of contextual information, and our method for analyzing words in contexts using independent component analysis. We also discuss our methods for the analysis of the results.

2.1 Data

We selected the English-Finnish pair from the sentence-aligned Europarl parallel corpus of the European Parliament proceedings (Koehn, 2005) for our initial experiments. The texts were preprocessed by replacing all characters with their lowercase versions, numbers with a special symbol and by removing non-word codes and punctuation marks. In the corpus, there were $602,153$ aligned regions, where there were 25 tokens per regions in average for English and 20 for Finnish. There were a total of $15,215,650$ tokens (words in running text) and $50,111$ types (unique word forms) in the English text and $10,437,070$ tokens and $350,972$ different types in the Finnish text.

We began by concatenating the aligned regions for English and Finnish together. This defined the contexts according to the bag of words model, where the word order is not considered. For both languages, we selected $10,000$ most frequent types to be analyzed in the contexts of $1,000$ most frequent types in both languages. After removal of duplicate types, we collected the frequencies of co-occurrences of each word j with the context words in the concatenated regions as a vector $\mathbf{x}_j$, which formed columns of a context-word co-occurrence frequency matrix $\mathbf{X}$ of size $1,983 \times 19,758$. The logarithm of the elements of the matrix $\mathbf{X}$ added by one was taken to dampen the effect of high frequency differences.

The method above differs from the approaches in Ritter and Kohonen (1989) and Honkela et al. (1995) because of the nature of the parallel corpus. For instance, Honkela et al. (1995) calculated contexts for two adjacent words in English text. For sentence aligned corpus, however, there is no match between tokens in across languages inside the aligned regions. Naturally, a more refined alignment would make this possible. It seems reasonable to assume that the use of longer contexts contains information more on the semantic level than the syntactic level, whereas the syntactic level information could be gathered using shorter contexts.

2.2 Independent component analysis

In the classic version of the linear, instantaneous ICA model (Comon, 1994; Hyvärinen et al., 2001), each observed random variable $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ is represented as a weighted sum of independent random variables $\mathbf{s} = (s_1, \ldots, s_k, \ldots, s_n)^T$

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{1}$$

where mixing matrix $\mathbf{A}$ contains the weights which are assumed to be different for each observed variable. Both the mixing matrix $\mathbf{A}$ and the independent components (features) $\mathbf{s}$ are learned in an unsupervised manner from the observed data $\mathbf{x}$. It should be noted that the ICA model leaves the number of components, the order of the components and scale and sign of the components ambiguous.

For our analysis of the observed context-word frequencies, we applied FastICA (Hyvärinen, 1999) algorithm with the standard maximum-likelihood estimation by setting the nonlinearity to the $\tanh$ function, and using symmetric orthogonalization. The dimension of the data was first reduced to 100 by principal component analysis (PCA) in order to decorrelate the data, to reduce overlearning and to get a square mixing matrix $\mathbf{A}$. From the whitened data, i.e., after PCA and normalization of variances, 100 components were extracted with ICA.

The resulting components $\mathbf{s} = (s_1, \ldots, s_n)^T$ create a feature representation in the component space, which makes it possible to analyze how the positions of the words in the two languages are related. Moreover, each component can be analyzed separately by considering which words are most prominent, i.e., have the highest (absolute) value of that component. It seems that the estimated components s_k are skewed in nature. The sign ambiguity of Eq. 1 allows us to have only positively skewed components. Now we need only to consider high positive values, which makes the analysis of the components more simple.

The word vectors $\mathbf{s}_j$ are produced as a resulta of the independent component analysis of the sum of contexts $\mathbf{x}_j$ of the word j. The word vectors $\mathbf{s}_j$ can be analyzed as a bilingual lexicon for the words in the two languages, where each component s_k encodes some interesting feature. This results in a two-fold analysis that is discussed in more detail in the following.

2.3 Comparing word similarities

We can consider the whole component space $\mathbf{s} = (s_1, \ldots, s_k, \ldots, s_n)^T$ and see which words are close to each other. An ideal model would be that translation of a word would be found as the closest point in the space. In addition to that, related words would be closer to each other than unrelated words. This is what latent semantic analysis (LSA) does (Deerwester et al., 1990). We measure similarity between the query word vector $\mathbf{s}_1$ and any other word vector $\mathbf{s}_2$ as the cosine of the angles between the vectors

$$d_{\cos}(\mathbf{s}_1, \mathbf{s}_2) = \frac{\mathbf{s}_1^T \mathbf{s}_2}{\|\mathbf{s}_1\| \|\mathbf{s}_2\|} \tag{2}$$

where the value $d_{\cos}$ is in the range $[-1, 1]$. The highest value $+1$ is obtained for the query word vector itself.

2.4 Examination of the components

Each component s_k can be thought as a semantically interesting feature, which has related words as the most prominent words and for the rest of the words the component has a value near zero, i.e., we consider the values s_k given for each word and list the corresponding words in descending order.

Table 1: Closest words in the ICA space to 'suomi' (left) and 'finland' (right)

suomi	$d_{\cos}$	finland	$d_{\cos}$
suomi	1.00	finland	1.00
finland	0.82	suomen	0.83
itävalta	0.76	suomi	0.82
finnish	0.74	sweden	0.79
britannia	0.74	suomessa	0.77
saksa	0.73	austria	0.73
...		...	

3 Experiments and results

Preliminary experiments using a bilingual parallel corpus are discussed here. Contextual data was collected into data matrix $\mathbf{X}$ and 100 components were extracted with the FastICA algorithm as reported above. The emergent components define a bilingual lexicon and each component can be analyzed separately.

We analyzed how well the found component space brings translations of words together. Table 1 shows the closest words for the Finnish word 'suomi' (left) and its English translation 'finland' (right) according to Eq. 2. The closest match is an exact translation of the query word. Considering all the closest matches, there is a clear pattern of country names in both languages: They are clearly related to the word we are comparing it to. It should be mentioned, that the word 'suomi' is in fact ambiguous having the meaning of both the name of the country (written with a capital S) and the name of the language. One can see that the English word meaning Finnish language is third on the list.

We repeated the experiment for 14 EU member states names both in English and in Finnish from the time of the recording of the proceedings (years 1996–2003). United Kingdom was excluded because it consists of two separate words. The results were similar to those described above. For member states in English, the corresponding word in some inflected form in Finnish was the closest word form, except for one case, where a possible translation was the second closest word form. The problem seemed to be more difficult to the other direction. Now five of the translations were found from the second closest word form and one translation was the sixth closest word form.

As an additional short analysis of how well our system performs, we took the 30 most frequent nouns from both the English and the Finnish corpus and checked whether our method could find a corresponding translation for them. As most common words in other word classes often correspond to function words and do not necessarily have direct one-word translation, they were left out of this short analysis. No stemming of the words was performed: The query words included inflective forms as well and a translated form was considered a match regardless of its inflection. The translations were checked manually using NetMot electronic dictionary from Kielikone Ltd.

We checked whether a corresponding term in the other language was found within four closest matching words (that could be in either language). Precision, $precision = C/S$, where C is the number of correct translations and S is the size of the source vocabulary, was used as an evaluation metric.

When Finnish query words were used, the translation precision was 0.67 when only the closest word was examined. The precision rose to 0.77, if two closest words were taken into account and to 0.83 when three or four closest words were considered. Finnish is a more compounding language than English, which is why there were Finnish query words that would be translated into two words in English. An example of this is 'jäsenvaltiot' for which the appropriate translation is 'member states'. Our method could find parts of collocations like this, but in this experiment it was considered a failure. If partial matches like this were taken into account, the precision was 0.96 when four closest words were considered.

For English query words, the precision was 0.53 for the closest word, 0.70 when using the two closest and 0.73 and 0.77 with three and four closest matches, respectively. If partial matches were considered as described above, the precision rose to 0.83 when four closest words were examined.

Compared to the precisions reported in Sahlgren and Karlgren (2005), our initial results with high-frequency words are a little worse than theirs, but on the same level as their reported precision over words that occur more than 100 times. It should be mentioned that Sahlgren and Karlgren (2005) utilized lemmatization and the language pairs, context type and alignment level differed from the ones used here.

We can also study the individual components s_k obtained as they are interesting themselves. As an example we present Table 2 showing the most prominent words for selected three components. It can be seen that the components list words in both languages related to 1) countries, languages and nationalities, 2) values, and 3) differences. The relations between the listed words include, for instance, inflected word forms: 'arvoja' (plural partitive of 'value'), 'arvojen' (plural genitive of 'value'); words used together or close to each other: 'eroja' (plural partitive of 'difference'), 'välillä' ('between'); translations: 'france', 'ranska'; or otherwise closely related words: 'france', 'french'.

Table 2: Most prominent words for three example components (columns) that list clearly related words in both languages

saksan	values	eroja
ranskan	rauhan	different
germany	demokratian	difference
france	vapauden	välillä
french	democracy	erilaista
german	ihmisoikeuksien	differences
sweden	arvoja	erot
netherlands	solidarity	toisiaan
ranska	peace	disparities
belgian	arvojen	eri
ruotsin	kunnioittaminen	erilaiset
saksa	oikeusvaltion	differ
italian	principles	differing
kingdom	continent	eroavat
. . .	. . .	. . .

The small number of extracted components is naturally not able to produce a unique concept for everything. Instead, the components model the concepts using a sparse bag of concepts model, where only a few of the features are active for each concept.

4 Discussion

Our initial experiments covered only a bilingual case, but the method is directly applicable to a multilingual case if a parallel multilingual corpus is available. Further research will include experiments with the multilingual material using more than two languages as well as a more thorough analysis of the results.

In our experiments, we showed briefly that the analysis of the components produced by ICA can be interesting. As mentioned earlier, using shorter contexts, one is able to obtain ICA-based features that are more syntactic in nature, whereas more semantic features can be found when the context is sufficiently large.

In addition to simply finding the closest matches based on the vector representation, the component analysis can be seen as a tool to dig a little deeper in the level of meaning of words: What do the words (or their underlying meanings) have in common? The use of multilingual corpora gives us a possibility to analyze whether the individual components are meaningful themselves. For instance, we can see what kind of components are active for words in different languages with the same or similar meaning.

Gärdenfors (2000) presents the conceptual spaces model for modeling conceptual representations in a cognitive framework. In that model, the concepts are seen as areas in a multi-dimensional conceptual space built using different quality dimensions (of which the simplest examples could be weight, width, height, or temperature). Our viewpoint is that we are able to obtain representations using the emergent ICA components that are in accord with the conceptual spaces model. Undeterministic statistical analysis of textual data could provide features that align with human perception of the world. Our research focus will be in deeper analysis and understanding of the emergent features.

5 Conclusions

Contextual information for words was analyzed by independent component analysis to produce a component space for words. Our hypothesis was that the obtained components create a meaningful representation instead of a latent representation produced by, for instance, random indexing. We tested our assumption with a bilingual parallel corpus, which enabled us to examine both the found component space and the individual components.

A preliminary experiment utilized a sentence aligned parallel corpora of English-Finnish proceedings of the European parliament. The components obtained form a space in which related words in both languages appear closer to each other than unrelated words. This makes it possible to find translation candidates for words by comparing the similarities of the word vectors in the component space. Moreover, the components clearly encode semantic features using both languages.

We conclude, that the results reported in this paper support our assumption that independent component analysis is able to find components that are meaningful and a good feature representation for words.

References

Pierre Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314, 1994.

Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of American Society of Information Science*, 41(6):391–407, 1990.

Peter Gärdenfors. *Conceptual Spaces: The Geometry of Thought*. MIT Press, 2000.

Timo Honkela, Ville Pulkki, and Teuvo Kohonen. Contextual relations of words in Grimm tales analyzed by self-organizing map. In *Proceedings of ICANN-95*, pages 3–7, 1995.

Aapo Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent Component Analysis*. John Wiley & Sons, 2001.

Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT Summit X*, 2005.

Helge Ritter and Teuvo Kohonen. Self-organizing semantic maps. *Biological Cybernetics*, 61(4):241–254, 1989.

Magnus Sahlgren and Jussi Karlgren. Automatic bilingual lexicon acquisition using random indexing of parallel corpora. *Natural Language Engineering, Special Issue on Parallel Texts*, 11(3): 327–341, 2005.

Ensembles of nearest neighbour classifiers and serial analysis of gene expression

Oleg Okun[⋆] Helen Priisalu[†]

[⋆]University of Oulu, FINLAND [†]Tallinn University of Technology, ESTONIA

Abstract

In this paper, we represent experimental results obtained with ensembles of nearest neighbour classifiers on the binary classification problem of cancer classification using serial analysis of gene expression (SAGE) data. Nearest neighbours are selected as classifiers since they were rarely employed in building ensembles because their predictions are stable to small perturbations of data, which does not make individual classifiers diverse. To remedy this problem, four feature selection methods, all belonging to the filter model, are applied prior to classification. In addition, diversity-based pruning of classifiers is utilised before combining them into an ensemble by means of three combination methods (majority vote, weighted majority vote, and Naïve Bayes combination with a correction for zeroes). Our preliminary conclusions demonstrate that the best results in terms of error rate, false positive rate, true positive rate, and their combinations are achieved with the Naïve Bayes combination of the pruned classifiers, where each classifier works with a subset of the original genes.

1 Introduction

Cancer is one of the three leading causes of death in developed countries. Cancers are caused by cells which grow progressively without any regulation. Curing cancer means destruction of these cells, which, in turn, implies that the immune system should identify tumour cells and healthy cells and generate an immune response against the tumour cells. Hence, there is a great challenge and oppotunity for computer science to contribute to biological research on cancer and associated biomarkers.

However, cancer tumours are not easily detectable by the immune system because of their genetic instability. For example, cancer cells can “deceive” the immune system so that it is unable to kill them. Therefore for computers it is also a challenge to discriminate between healthy and tumour cells. Besides, there are thousands of genes for only several dozens of samples taken from patients. This causes the curse of dimensionality to occur that negatively influences on classification accuracy.

Many methods for cancer prediction have been published (see (Dudoit and Fridlyand, 2003; Lu and Han, 2003) for the comprehensive review). The vast majority of them apply a single classifier (nearest neighbour, support vector machine, neural network, decision tree) in order to solve the problem. Because none of classifiers is perfect and superior to the others, their ensembles are thought to explore a wider solution space, thus avoiding local optima of the individual classifiers and yielding higher accuracy. Ensembles are especially suitable in cancer research where classification has to be done based on few biological samples and a multitude of genes: it is a hard problem for any single classifier to capture true hypothesis from this kind of data.

In general, it is well known that an ensemble can provide higher accuracy than a single best classifier if its members are sufficiently diverse and accurate. Diversity usually means that classifiers make different predictions, i.e., they do not make errors on the same samples. One of the most typical ways to make classifiers diverse is to let each classifier to work only with a certain subset of the original features so that these subsets can overlap but they do not totally coincide.

Our goal in this paper is not just ensembles of classifiers, but ensembles of *nearest neighbour* (NN) classifiers. The following reasons motivated us in this choice:

- NNs already demonstrated very good performance in several bioinformatics tasks (Dudoit and Fridlyand, 2003; Okun and Priisalu, 2006a);
- NNs provide fast classification since the number of samples is typically less than 100 for gene expression data;

- NNs do not require any parameters to tune (k can be fixed, for example, to 1, 3, and 5);
- there is not much work on ensembles of NNs because NNs are stable classifiers (regarding small perturbations in the training set) that are difficult to combine by the traditional bagging, boosting or error-correcting output coding (see the state-of-the-art in (Okun and Priisalu, 2005));
- applications of ensembles of NNs for gene expression analysis are rare due to the above mentioned challenge.

Diversity among selected features decorrelates errors made by individual NN classifiers and it can overcome the stability of NNs.

2 Feature selection methods

Feature selection methods are divided into filters and wrappers. Wrappers rely on a classifier in order to judge on feature importance while filters do not. Independently of this fact, methods of both types typically select one feature at a time, thus ignoring interaction between features. There are known few feature selection methods targeting feature interaction and among them is pairwise feature selection proposed in (Bø and Jonassen, 2002). We will utilise this method as well as two one-feature-at-a-time methods employed in (Bø and Jonassen, 2002) for gene selection. All the methods expect *two-class* data.

2.1 Method of Bø and Jonassen

This filtering method evaluates how well a pair of genes in combination distinguishes two classes. First, each pair is evaluated by computing the projections of the training data on the diagonal linear discriminant (DLD) axis when using only two genes constituting this pair. Then the two-sample t-statistic is computed on the projected points and assigned to a given pair as its score. Bø and Jonassen proposed two variants of gene selection based on pair scores, called ‘all pairs’ (exhaustive search) and ‘greedy pairs’ (greedy search).

The all-pairs variant targets all possible pairs of genes. First, all pairs are sorted in descending order of their score. After that, the pair with the highest pair score is selected and all other pairs containing either gene included in this pair are removed from the sorted list of pairs. Then the next highest-scoring pair is found from the remaining pairs and all other pairs containing either gene in this pair are removed from the list, and so on. This procedure terminates when the pre-specified number of genes is reached.

Since the all-pairs variant is computationally demanding, an alternative evaluating only a subset of all pairs is proposed (greedy pairs). The greedy-pairs variant first ranks all genes based on indivudual t-score. Next, the best gene g_i ranked by its t-score is selected. Among all other genes, the gene g_j that together with g_i maximises the pair t-score is found so that g_i and g_j form a pair. These two genes are then removed from the gene set and a search for the next pair is performed until the desired number of genes is selected.

2.2 Greedy forward selection

It is the greedy feature selection method, incrementally adding genes one-by-one; hence the name ‘forward selection’. First, individual t-score is computed for each gene (two-class data using a given gene are projected onto the DLD axis and the two-sample t-statistic on the projected points is taken as the gene score) and genes are ranked according to their t-score.

The first gene to be selected is the one with the highest t-score. Next, other genes are added so that adding a gene maximises the two-sample t-score of a new subset. The t-score at step i is computed on the data points projected on the DLD axis using the expression values of the i selected genes. Greedy forward selection stops when collecting the desired number of genes.

2.3 Individual ranking

Two previous methods belong to gene subset ranking techniques because each time a group of genes gets a rank (Lu and Han, 2003) in contrast to individual ranking which grades each gene in isolation from others. Compared to gene subset ranking, individual ranking looks handicapped since it ignores interactions among genes. However, it is the fastest method among the three, because it is the simplest one. To individually rank genes, it employes the two-sample t-statistic on the *expression values* of the genes across two classes. That is, the expression values of each gene for normal and cancer exemplars form two samples used in the statistical test returning t-score of that gene.

3 Base classifiers

As was mentioned, our interest is in nearest neighbour classifiers. In particular, we will utilise the weighted k-NN ($k = 1, 3, 5$). It is known that k-NN is sensitive to irrelevant features (Aha, 1992); hence, gene selection applied prior to classification will help to filter out irrelevant genes.

Weights are associated with the original genes. The ith weight is equal to the number of times the ith gene was selected during cross-validation of gene selection. It means that weights reflect importance or relevance of genes: the higher weight, the more relevant gene.

Given two n-dimensional patterns x and y, the similarity between them is determined by the following formula:

$$\sqrt{\sum_{i=1}^{n} w_i(x_i - y_i)^2} = \sqrt{\sum_{i=1}^{n}(\sqrt{w_i}x_i - \sqrt{w_i}y_i)^2} \ ,$$

where w_i is the weight of the ith gene. Thus, patterns have to be weighted before searching for nearest neighbours.

Four gene selection methods introduced in the previous section and three values of the number of nearest neighbours, k (1, 3, 5), yield twelve combinations summarised in Table 1. We call these combinations the base classifiers. Each base classifier carries out feature selection followed by classification using the selected features.

Table 1: Base classifiers and their IDs

ID	Feature selection	Classifier
1	Bø-Jonassen (greedy pairs)	1NN
2	Bø-Jonassen (greedy pairs)	3NN
3	Bø-Jonassen (greedy pairs)	5NN
4	Bø-Jonassen (all pairs)	1NN
5	Bø-Jonassen (all pairs)	3NN
6	Bø-Jonassen (all pairs)	5NN
7	Greedy forward selection	1NN
8	Greedy forward selection	3NN
9	Greedy forward selection	5NN
10	Individual ranking	1NN
11	Individual ranking	3NN
12	Individual ranking	5NN

4 Pruning of the base classifiers

It is imperative that in order for an ensemble to outperform a single best classifier, the former must be composed of diverse classifiers that are also sufficiently accurate at the same time (Kuncheva, 2004). In many models injecting diversity into an ensemble, e.g., via random sampling of input data like in (Breiman, 1996) or via random feature sampling like in (Bay, 1999), the diversity is not explicitly measured and controlled. Hence, it is difficult to judge how diverse the base classifiers and this appeals to incorporate some *explicit* measure of diversity into selection of the base classifiers.

Though all the base classifiers are typically combined into an ensemble, it is possible to select only certain classifiers for fusion. This operation is called pruning of the base classifiers. Prodromidis et al. (1998) proposed to use diversity-based pruning before classifier combination. They defined diversity for a pair of classifiers D_i and D_j as the number of patterns where D_i and D_j produce different predictions. Their diversity-based algorithm is given in Table 2, where $\mathcal{H}$ and $\mathcal{C}$ are sets containing the original and selected base classifiers, respectively.

Table 2: Diversity-based pruning algorithm (Prodromidis et al., 1998)

```
**Let** $\mathcal{C} = \emptyset$, $M :=$ maximum number of classifiers
**For** $i := 1, \ldots, \mid \mathcal{H} \mid -1$ **do**
    **For** $j := i, \ldots, \mid \mathcal{H} \mid$ **do**
        **Let** $z_{i,j}$ :=number of patterns where
            $D_i$ and $D_j$ give different predictions
**Let** $D' :=$ the classifier with the highest accuracy
$\mathcal{C} := \mathcal{C} \cup D'$, $\mathcal{H} := \mathcal{H} - D'$
**For** $i := 1, \ldots, M - 1$ **do**
    **For** $j := 1, \ldots, \mid \mathcal{H} \mid$ **do**
        **Let** $S_j := \sum_{k=1}^{|\mathcal{C}|} z_{j,k}$
    **Let** $D' :=$ the classifier from $\mathcal{H}$ with
            the highest $S_j$
    $\mathcal{C} := \mathcal{C} \cup D'$, $\mathcal{H} := \mathcal{H} - D'$
```

First, the most accurate classifier is included into $\mathcal{C}$. Next, in each round, the algorithm adds to the list of selected classifiers the classifier D_k that is most diverse to the classifiers chosen so far, i.e., the D_k that maximises the diversity S over $\mathcal{C} \cup D_k, \forall k \in \{1, 2, \ldots, |\mathcal{H}|\}$. The selection ends when the M most diverse classifiers are selected[1].

Prodromidis et al. (1998) showed on two data sets of real credit card transactions that diversity-based pruning can lead to the ensemble with higher accuracy than that combining all classifiers. That is why

[1] As more base classifiers are included into $\mathcal{C}$, the diversity of the set of classifiers decreases. Hence M is a threshold that prevents redundant classifiers to be selected in the ensemble.

we believe that pruning of classifiers can improve ensemble performance in our case, too.

5 Combination of the base classifiers into an ensemble

We employ three classifier fusion methods: majority vote, weighted majority vote, and Naïve Bayes. All of them combine class labels of the base classifiers. Notation and formulae in this section are borrowed from (Kuncheva, 2004).

Let us assume that there are L classifiers D_i, $i = 1, \ldots, L$ and c different classes. The label outputs of these classifiers compose c-dimensional binary vectors $[d_{i,1}, \ldots, d_{i,c}] \in \{0,1\}^c$, $i = 1, \ldots, L$, where $d_{i,j} = 1$ if D_i labels the input $\mathbf{x}$ in class ω_j, and $d_{i,j} = 0$ otherwise.

5.1 Majority vote

The majority vote results in an ensemble decision for class ω_k if $\sum_{i=1}^{L} d_{i,k} = \max_{j=1}^{c} \sum_{i=1}^{L} d_{i,j}$, i.e., class ω_k got more votes than any other class. Ties are randomly resolved. In the case of two classes, the winning class should get 50 percent of the votes +1.

5.2 Weighted majority vote

If accuracies of the base classifiers in the ensemble vary, then it is reasonable to give the more accurate classifiers larger weight in making the final decision. Given that the label outputs $d_{i,j}$ provide degrees of support for the classes as above, the discriminant function for class ω_j is obtained by weighted majority vote as $g_j(\mathbf{x}) = \sum_{i=1}^{L} b_i d_{i,j}$, where $b_i = \log \frac{p_i}{1-p_i}$[2] is a weight for classifier D_i, and p_i is accuracy of D_i. Since $d_{i,j}$ can be either 0 or 1, the value of the discriminant function will be the sum of the coefficients of those base classifiers whose output for $\mathbf{x}$ is ω_j.

5.3 Naïve Bayes

This combination method assumes that individual classifiers are mutually independent; hence the name 'naïve'. Let $s_1, \ldots, s_L$ be crisp class labels assigned to $\mathbf{x}$ by classifiers $D_1, \ldots, D_L$, respectively. The independence assumption means that the support for class ω_k is $\mu_k(\mathbf{x}) \propto P(\omega_k) \prod_{i=1}^{L} P(\omega_k \mid s_i)$, where $P(\omega_k \mid s_i)$ is the probability that the true class is ω_k given that D_i assigns $\mathbf{x}$ to class s_i. For each classifier D_i, a $c \times c$ confusion matrix CM^i is calculated by applying D_i to the training data set. The (k,s)th entry of this matrix, $cm^i_{k,s}$ is the number of patterns whose true class label was ω_k, but they were classified in class ω_s by D_i. By N_k we denote the total number of patterns from class ω_k. Taking $cm^i_{k,s_i}/N_k$ as an estimate of the probability $P(\omega_k \mid s_i)$, and N_k/N ($N = \sum_{j=1}^{c} N_j$) as an estimate of the prior probability for class ω_k, we obtain that $\mu_k(\mathbf{x}) \propto \frac{1}{N N_k^{L-1}} \prod_{i=1}^{L} cm^i_{k,s_i}$.

To avoid nullifying $\mu_k(\mathbf{x})$ regardless of other estimates when an estimate of $P(\omega_k \mid s_i) = 0$, the last formula can be rewritten as $\mu_k(\mathbf{x}) \propto \frac{N_k}{N} \left\{ \prod_{i=1}^{L} \frac{cm^i_{k,s_i} + 1/c}{N_k + 1} \right\}^B$, where B is a constant (we always set it to 1 in experiments).

6 Data set

We chose the data set containing the expressions of 822 genes in 74 samples. Its description can be found in (Gandrillon, 2004). The distribution of normal and cancerous samples is imbalanced with the bias toward the latter (24 samples are normal while 50 samples are cancerous). Unlike many other data sets with one or few types of cancer, the one we are going to work with contains 9 different types of cancer, which means that there are only 5 samples per cancer type, on average. Hence, we opted to ignore the difference between cancer types and to treat all cancerous samples as belonging to a single class. This reduced our classification task to two classes (either 'normal' or 'cancer').

This data set was produced according to the Serial Analysis of Gene Expression (SAGE) technique (Velculescu et al., 1995; SAGENET), which is designed to take advantage of sequencing technology. It was originally conceived for use in cancer studies. SAGE provides a statistical description of the mRNA population present in a cell without prior selection of the genes to be studied (Aldaz, 2003). This is the main advantage of SAGE over microarray approaches (cDNA and oligonucleotide microarrays) that are limited to the genes represented in the chip.

SAGE "counts" the number of transcripts or tags for each gene, where the tags substitute the expression levels used in microarray approaches. As a result, counting sequence tags yields *positive integer* numbers in contrast to microarray measurements. Machine learning methods can be applied to tag counts in order to determine highly expressed genes.

[2] In (Kuncheva, 2004) it is showed that such a definition of b_i maximises the accuracy of the ensemble when the outputs of the base classifiers are combined with the weighted majority vote.

For example, a normal tissue sample can be compared against a cancer tumour to determine which genes tend to be more (or less) active. Besides, the SAGE data can be directly compared with other data generated from a different laboratory.

In SAGE, the mRNA sequences do not need to be known a priori so that unknown genes can be discovered. In other words, it is precisely in the area of identification of novel tumour markers (for all tumour types) where SAGE has been demonstrated to be the most useful (see (Aldaz, 2003) for details). However, microarray experiments are much cheaper to perform. In addition, small length of the tags makes them susceptible to non-specific interactions, and errors resulting from the use of sequencing technology may substantially degrade the correct assignment of a tag count to a gene. Finally, SAGE is not very suitable for the comparison of large samples (e.g., several hundreds) in a relatively short time, in contrast to microarray approaches. Nevertheless, currently there is no absolutely perfect technology.

7 Experiments

Based on the overview of the ensembles used for cancer classification[3], we can say that none of the papers utilised SAGE data in cancer research. The Naïve Bayes combination rule was rarely applied in contast to majority vote. Diversity-based pruning of the base classifiers was not employed, except for (Hong and Cho, 2005). Feature selection if used was simple and often ignoring gene interactions. In combination with the well-known fact that bagging and boosting do not improve performance of nearest neighbour classifiers, all these observations leave a lot of space for research that is reported below.

First, we performed two-class discrimination in the original, high dimensional space of 822 genes. Error rates for three NN classifiers are shown in Table 3 when using leave-one-out cross-validation (LOOCV). These results will serve for a comparison with those obtained in low dimensional gene selection-induced space.

Next, gene selection methods were applied, followed by classification as given in Table 1. For Bø and Jonassen's method, the number of attributes to be selected was set to 2, 4, . . . , 400, while for other two methods it ranged from 1 to 400.

For each base classifier in Table 1, both gene selection and classification were cross-validated using leave-one-out cross-validation. Gene selection was first done for each CV fold separately. Though among the selected genes there were those shared by many or even all folds, we observed that the subsets of selected genes varied from fold to fold. However, from the biological point of view, it would be more convenient to analyse one subset of genes. Hence, we preserved only the most frequently selected genes in all folds and discarded all the other genes, i.e., we filtered out rare genes by treating them as noise[4]. To do so, a frequency of occurrence of each gene was calculated over all subsets of the selected genes and those genes whose frequency was less than the empirically chosen threshold thr were discarded[5]. The preserved genes were then used for cross-validation of a classifier.

Table 3: LOOCV error rates in the original space

Classifier	LOOCV error rate
1NN	0.2703
3NN	0.4054
5NN	0.3108

This procedure was carried out for each value of the number of genes to be selected and with each base classifier. For each base classifier, the number of genes leading to the lowest CV error was recorded. Later on, the base classifiers generated predictions using this optimal number of selected genes, k_{opt}. The actual number of genes, $n(k_{opt})$, used by different classifiers is shown in Table 4[6].

Before combining classifiers into an ensemble, classifier pruning occurred as described above. The predictions (class labels) of base classifiers remaining after pruning were finally fused by each of the three combination techniques (majority vote, weighted majority vote, and Naïve Bayes combination with a correction for zeroes).

7.1 Performance estimation of the base classifiers and their ensembles

To characterise the base classifiers and their ensembles, we utilised 5 measures: **ER** (LOOCV error rate), **TPR** (true positive rate or sensitivity), **FPR** (false positive rate or 1-specificity), **TPR-FPR**

[3] It is reported elsewhere.

[4] Rare genes were the key reason why we did not test a classifier on the left-out samples for each CV subset.

[5] The value of thr is naturally to relate to the number of CV folds, N_{folds}, e.g., thr is equal to ℓN_{folds}, where $0 < \ell \leq 1$.

[6] It is often that $n(k_{opt}) \neq k_{opt}$ for a given classifier because subsets of genes vary from fold to fold.

Table 4: The actual number of genes used by different base classifiers for two values of thr: 32 and 74. The total number of selected genes used by all the twelve base classifiers is 242 and 367, respectively

Classifier	$thr = 32$	$thr = 74$
1	227	191
2	28	4
3	4	4
4	180	176
5	18	12
6	8	11
7	7	98
8	6	25
9	6	10
10	210	268
11	19	4
12	38	4

spread, and **F-measure**. Table 5 explains the meaning of true positives, true negatives, false positives, and false negatives. ‘Positive’ and ‘negative’ stand for ‘cancer’ and ‘healthy’, respectively.

Table 5: Confusion matrix for the two-class classification problem. TP = the number of true positives; FN = the number of false negatives; FP = the number of false positives; TN = the number of true negatives

	Predicted	
True	cancer	normal
cancer	TP	FN
normal	FP	TN

The error rate (ER) is (FP+FN)/(TP+TN+FP+FN). The true positive rate (TPR) is the ratio of the number of true positives to the number of positives: TPR = TP/(TP + FN). The false positive rate (FPR) is the ratio of the number of false positives to the number of negatives: FPR = FP/(FP + TN). TPR or FPR are one-sided estimates. Hence other two measures complement them, which combine both TPR and FPR. The TPR-FPR spread is the difference of TPR and FPR (TPR-FPR). It characterises the ability of a classifier to make correct predictions while minimising false alarms. The F-measure borrowed from information retrieval combines precision and recall, where Precision = TP/(TP + FP) and Recall = TPR: F-measure = (2*Precision*Recall)/(Precision + Recall).

7.2 Results of experiments

Results for two values of thr (32 and 74) are summarised in Tables 6 and 7, respectively. It can be seen that they are much better than those obtained without gene selection. The following abbreviations are used: MV (majority vote), WMV (weighted majority vote), NB (Naïve Bayes). Figures in brackets after MV, WMV or NB stand for the maximal number of base classifiers in the ensemble, selected by pruning 12 original base classifiers, with ‘12’ implying no pruning. Characteristics of a single best classifier among the twelve are shown in italics. Thus, when $thr = 32$, the best classifier is the classifier no.7 while when $thr = 74$, the best classifier is the classifier no.1. Figures in the column ‘Classifier/Ensemble’ such as 1, 2, 3, $\ldots$, 12 without MV, WMV or NB mean the base classifiers (see Table 1). Best results for each of the five measures described in the previous subsection are shown in bold.

The base classifiers were added to the ensemble in the following order: 7, 11, 5, 1, 6, 3, 12, 9, 4, 8, 2 (for thr=32) and 1, 8, 3, 12, 5, 9, 10, 2, 7, 11, 6 (for thr=74). It seems that both variants of pairwise gene selection (Bø and Jonassen, 2002) were among the top selected, though the small number (12) of base classifiers prevents us from drawing comprehensive conclusions whether pairwise gene selection results in better performance than the one-gene-at-a-time methods. However, pairwise gene selection points to pairs of related genes and thus, it can facilitate the analysis of biological relevance of the selected genes.

Ensembles of classifiers were almost always better in terms of quantitative characteristics used than single best classifiers. Classifier pruning was especially valuable in getting better results. Among three combination methods, the Naïve Bayes fusion turned out to be best. Weighted majority vote was not superior to majority vote.

Regarding processing time, individual ranking was naturally much faster than other employed methods. Greedy forward selection was slower (at least two times) than either variant of the Bø-Jonassen algorithm. Leave-one-out cross-validation of gene selection turned out to be very time consuming for all methods but individual ranking. Hence, we recommend to apply k-fold cross-validation (k is equal either 5 or 10) instead in the future.

Finally, preliminary analysis of biological significance of our results is reported in the companion paper (Okun and Priisalu, 2006b) and therefore it is omitted here.

8 Conclusion

In this paper, we applied ensembles of nearest neighbour classifiers for cancer classification using the SAGE data set. Our experiments demonstrate that gene selection prior to NN classification makes the base classifiers diverse and accurate. Diversity-based pruning of the base classifiers before combining them into an ensemble dramatically improves the ensemble performance in terms of various characteristics such as error rate, false positive rate, true positive rate, and their combinations. The Naïve Bayes combination of the pruned base classifiers generally outperforms the standard majority vote and its weighted variant, though we advise to run several combination schemes since combination is fast.

Acknowledgements

Oleg Okun would like to thank the Riitta and Jorma J. Takanen Foundation (Finland) for the financial support of his work. Both authors are grateful to reviewers for valuable comments.

References

David W. Aha. Generalizing from case studies: a case study. In *Proceedings of the Ninth International Conference on Machine Learning, Aberdeen, Scotland*, pages 1–10, 1992.

Marcelo C. Aldaz. Serial analysis of gene expression (SAGE) in cancer research. In Marc Ladanyi and William L. Gerald, editors, *Expression Profiling of Human Tumors: Diagnostic and Research Applications*, pages 47–60. Humana Press, Totowa, New Jersey, 2003.

Stephen Bay. Nearest neighbor classification from multiple feature sets. *Intelligent Data Analysis*, 3 (3):191–209, 1999.

Trond H. Bø and Inge Jonassen. New feature selection procedures for classification of expression profiles. *Genome Biology*, 3(4):0017.1–0017.11, 2002.

Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

Sandrine Dudoit and Jane Fridlyand. Classification in microarray experiments. In Terry Speed, editor, *Statistical Analysis of Gene Expression Microarray Data*, Interdisciplinary Statistics, pages 93–158. Chapman & Hall/CRC, Boca Raton, Florida, 2003.

Olivier Gandrillon. Guide to the gene expression data. In *Proceedings of the ECML/PKDD Discovery Challenge Workshop, Pisa, Italy*, pages 116–120, 2004.

Jin-Hyuk Hong and Sung-Bae Cho. Cancer prediction using diversity-based ensemble genetic programming. In *Proceedings of the Second International Conference on Modeling Decisions for Artificial Intelligence, Tsukuba, Japan*, pages 294–304, 2005.

Ludmila I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. John Wiley & Sons, Inc., Hoboken, New Jersey, 2004.

Ying Lu and Jiawei Han. Cancer classification using gene expression data. *Information Systems*, 28(4): 243–268, 2003.

Oleg Okun and Helen Priisalu. Multiple views in ensembles of nearest neighbor classifiers. In *Proceedings of the ICML Workshop on Learning with Multiple Views (in conjunction with the 22nd International Conference on Machine Learning), Bonn, Germany*, pages 51–58, 2005.

Oleg Okun and Helen Priisalu. Fast nonnegative matrix factorization and its application for protein fold recognition. *EURASIP Journal on Applied Signal Processing*, 2006:Article ID 71817, 8 pages, 2006a.

Oleg Okun and Helen Priisalu. Multi-class cancer classification using ensembles of classifiers: Preliminary results. In *Proceedings of the Workshop on Probabilistic Modeling and Machine Learning in Structural and Systems Biology, Tuusula, Finland*, pages 137–142, 2006b.

Andreas L. Prodromidis, Salvatore Stolfo, and Philip K. Chan. Pruning classifiers in a distributed meta-learning system. In *Proceedings of the First National Conference on New Information Technologies, Athens, Greece*, pages 151–160, 1998.

SAGENET. URL `http://www.sagenet.org`.

Victor E. Velculescu, Lin Zhang, Bert Vogelstein, and Kenneth W. Kinzler. Serial analysis of gene expression. *Science*, 270(5235):484–487, 1995.

Classifier/Ensemble	ER	TPR	FPR	TPR-FPR	F-measure	Ensemble	ER	TPR	FPR	TPR-FPR	F-measure
1	0.1757	0.8600	0.2500	0.6100	0.8687	MV (11)	0.0946	0.9600	0.2083	0.7517	0.9320
2	0.2027	0.9000	0.4167	0.4833	0.8571	WMV (11)	0.0946	0.9600	0.2083	0.7517	0.9320
3	0.1892	0.9000	0.3750	0.5250	0.8654	NB (11)	0.0676	0.9400	**0.0833**	0.8567	0.9495
4	0.1622	0.8600	0.2083	0.6517	0.8776	MV (9)	0.0811	0.9600	0.1667	0.7933	0.9412
5	0.2297	0.9000	0.5000	0.4000	0.8411	WMV (9)	0.0811	0.9600	0.1667	0.7933	0.9412
6	0.2297	0.8800	0.4583	0.4217	0.8381	NB (9)	0.0676	0.9600	0.1250	0.8350	0.9505
7	*0.1081*	*0.9000*	*0.1250*	*0.7750*	*0.9184*	MV (7)	0.1081	0.9600	0.2500	0.7100	0.9231
8	0.1081	0.9400	0.2083	0.7317	0.9216	WMV (7)	0.1081	0.9600	0.2500	0.7100	0.9231
9	0.1486	0.9000	0.2500	0.6500	0.8911	NB (7)	0.0676	0.9600	0.1250	0.8350	0.9505
10	0.1757	0.8600	0.2500	0.6100	0.8687	MV (5)	0.1081	**0.9800**	0.2917	0.6883	0.9245
11	0.2162	0.8600	0.3750	0.4850	0.8431	WMV (5)	0.1081	**0.9800**	0.2917	0.6883	0.9245
12	0.2432	0.8200	0.3750	0.4450	0.8200	NB (5)	**0.0541**	0.9600	**0.0833**	**0.8767**	**0.9600**
MV (12)	0.0811	0.9400	0.1250	0.8150	0.9400	MV (3)	0.0946	**0.9800**	0.2500	0.7300	0.9333
WMV (12)	0.0676	0.9600	0.1250	0.8350	0.9505	WMV (3)	0.0946	**0.9800**	0.2500	0.7300	0.9333
NB (12)	0.0811	0.9400	0.1250	0.8150	0.9400	NB (3)	0.0946	**0.9800**	0.2500	0.7300	0.9333

Table 6: Summary of results when thr =32

Classifier/Ensemble	ER	TPR	FPR	TPR-FPR	F-measure	Ensemble	ER	TPR	FPR	TPR-FPR	F-measure
1	*0.1892*	*0.8600*	*0.2917*	*0.5683*	*0.8600*	MV (11)	0.1486	0.9800	0.4167	0.5633	0.8991
2	0.2027	0.9000	0.4167	0.4833	0.8571	WNV (11)	0.1486	0.9800	0.4167	0.5633	0.8991
3	0.2027	0.8800	0.3750	0.5050	0.8544	NB (11)	0.0946	0.9000	**0.0833**	**0.8167**	0.9278
4	0.1892	0.8600	0.2917	0.5683	0.8600	MV (9)	0.1081	0.9800	0.2917	0.6883	0.9245
5	0.2432	0.8800	0.5000	0.3800	0.8302	WMV (9)	0.1081	0.9800	0.2917	0.6883	0.9245
6	0.2432	0.8600	0.4583	0.4017	0.8269	NB (9)	0.1351	0.8600	0.1250	0.7350	0.8958
7	0.2838	0.8200	0.5000	0.3200	0.7961	MV (7)	0.1216	0.9800	0.3333	0.6467	0.9159
8	0.2973	0.9000	0.7083	0.1917	0.8036	WMV (7)	0.1216	0.9800	0.3333	0.6467	0.9159
9	0.2973	0.8600	0.6250	0.2350	0.7963	NB (7)	**0.0811**	0.9400	0.1250	0.8150	0.9400
10	0.1892	0.8400	0.2500	0.5900	0.8571	MV (5)	0.1351	**1.0000**	0.4167	0.5833	0.9091
11	0.2297	0.8800	0.4583	0.4217	0.8381	WMV (5)	0.1351	**1.0000**	0.4167	0.5833	0.9091
12	0.2432	0.8800	0.5000	0.3800	0.8302	NB (5)	**0.0811**	0.9600	0.1667	0.7933	**0.9412**
MV (12)	0.1486	0.9600	0.3750	0.5850	0.8972	MV (3)	0.1351	0.9400	0.2917	0.6483	0.9038
WMV (12)	0.1081	0.9600	0.2500	0.7100	0.9231	WMV (3)	0.1351	0.9400	0.2917	0.6483	0.9038
NB (12)	0.1216	0.9000	0.1667	0.7333	0.9091	NB (3)	0.1351	0.9400	0.2917	0.6483	0.9038

Table 7: Summary of results when thr =74

On the Realization of Asymmetric High Radix Signed Digital Adder using Neural Network

Md. Sumon Shahriar†
Dept of Computer and Information science
University of South Australia
sumon_shahriar@yahoo.com

Tofail Ahammad
†Dept of Computer Science and Engg.
BRAC University, Dhaka, Bangladesh
tofailahammad@yahoo.com

Hafiz Md Hasan Babu
Dept of Comp. Sc. and Engg.
University of Dhaka,Bangladesh
hafizbabu@hotmail.com

Abstract

This paper presents an asymmetric high-radix signed-digital (AHSD) adder that performs addition on the basis of neural network (NN) and also shows that the AHSD number system supports carry-free (CF) addition by using NN. Besides, NN implies the simple construction in high-speed operation. The signed-digit number system represent the binary numbers that uses only one redundant digit for any radix $r \geq 2$, the high-speed adder in the processor can be realized in the signed-digit system without a delay of the carry propagation. A Novel NN design has been constructed for CF adder based on the AHSD $_{(4)}$ number system is also presented. Moreover, if the radix is specified as $r = 2^m$, where m is any positive integer, the binary-to-AHSD$_{(r)}$ conversion can be done in constant time regardless of the word-length. Hence, the AHSD-to-binary conversion dominates the performance of an AHSD based arithmetic system. In order to investigate how the AHSD number system based on NN design achieves its functions, computer simulations for key circuits of conversion from binary to AHSD $_{(4)}$ based arithmetic systems are made. The result shows the proposed NN design can perform the operations in higher speed than existing CF addition for AHSD.

1 Introduction

Addition is the most important and frequently used arithmetic operation in computer systems. Generally, a few methods can be used to speed up the addition operation. One is by using neural network design convert the operands from the binary number system to a redundant number system, e.g., the signed-digit number system [2][3] or the residue number system, so that the addition becomes carry-free (CF). This Neural Network (NN) design implies fast addition can be done, at the expense of conversion between the binary number system and the redundant number system. In this paper we focus on exploring high radix signed-digit (SD) numbers and we also present the *asymmetric high-radix signed-digit* (AHSD) number system using NN [4][5]. The idea of AHSD is not new. Instead of proposing a new number representation, our purpose is to explore the inherent CF property of AHSD by using NN. The CF addition in AHSD based on NN is the basis for our high-speed addition circuits. The conversion of AHSD to and from binary will be discussed in detail. By choosing $r = 2^m$, where m is any positive integer, a binary number can be converted to its canonical AHSD representation in constant time. We will also present one simple algorithm for converting binary bit pattern to make pairs in AHSD for radix-r. NN design for converting AHSD numbers to binary: the first stresses high speed and the other provide hardware reusability. Since the conversion from AHSD to binary has been considered the bottleneck of AHSD-based arithmetic computation based on NN, these NN design greatly improve the performance of AHSD systems. For illustration, we will discuss in detail the example on AHSD $_{(4),}$ i.e., the radix-4 AHSD number system. The proposed approach is practical thanks to the simple conversion.

2 Basic definitions

Here, few basic definitions will be discussed.

2.1 Neural Network

A neural network is a powerful data-modelling tool that is able to capture and represent complex input/output relationships. The motivation for the development of neural network technology stemmed from the desire to develop an artificial system that could perform "intelligent" tasks similar to those performed by the human brain. Neural networks resemble the human brain in the following two ways:

1. A neural network acquires knowledge through learning.
2. A neural network's knowledge is stored within inter-neuron connection strengths known as synaptic weights.

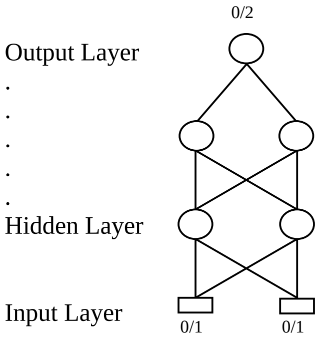


Figure 1: Neural Network prototype for AHSD number system addition.

In the figure, the simple prototype for NN is shown.

2.2 Asymmetric Number System [1][3]

The radix-*r* *asymmetric high-radix signed-digit* (AHSD) number system, denoted AHSD(*r*), is a positional weighted number system with the digit set $S_r = \{-1, 0, \ldots, r-1\}$, where $r > 1$. The AHSD number system is a minimally redundant system with only one redundant digit in the digit set. We will explore the inherent carry-free property in AHSD and develop systematic approaches for conversion of AHSD numbers from binary ones.

An *n*-digit number *X* in $\text{AHSD}_{(r)}$ is represented as

$X = (x_{n-1}, x_{n-2}, \ldots, x_0)_r$,

Where $x_i \in S_r$ for $i = 0, 1, \ldots, n-1$, and $S_r = \{-1, 0, 1, \ldots, r-1\}$ is the digit set of $\text{AHSD}_{(r)}$. The value of *X* can be represented as

$$X = \sum_{i=0}^{n-1} x_i r^i$$

Clearly, the range of *X* is $\left[\left(1-r^n\right)/\left(r-1\right), r^n - 1\right]$.

2.3 Binary to Asymmetric Number System conversion

Since the binary number system is the most widely used, we have to consider the conversion between the AHSD and binary number systems. Although the radix *r* may be any positive integer, simple binary-to-AHSD conversion can be achieved if $r = 2^m$ for any positive integer *m*. The reason for such simple conversion will be explained later. We assume $r = 2^m$ *in* what follows, unless otherwise specified. Note that there may be more than one $\text{AHSD}_{(r)}$ number representation for a binary number. For instance, the binary number $(0, 1, 1, 0, 0)_2$ can be converted to two different $\text{AHSD}_{(4)}$ numbers, i.e., $(1, -1, 0)_4$ and $(0, 3, 0)_4$. Hence, the binary-to-AHSD(*r*) conversion, being a one-to-many mapping, may be performed by several different methods. We would like to find an efficient and systematic conversion method that takes advantage of the carry-free property. Here we follow a general algorithm to make pairs to convert Binary to $\text{AHSD}_{(r)}$.

Algorithm for the conversion of AHSD from binary number system:

Step 1. Suppose given binary #bits=n
Step 2. If radix=2^m, where m is any positive integer then $2^p<m<2^{p+1}$ Where p=1,2,3……
Step 3. # Zero (0) will be padded in front of binary bits pattern 2^{p+1}-n
Step 4. Divide the array by m
Step 5. If each sub array is =m
Then stop
Step 6. Else
Divide each sub array by m

Proof:

Recurrence relation for the conversion from binary-to- AHSD numbers system is:

$$T(n)=\begin{cases} c & \text{if } n=m \text{ where c is a constant \& } m>2 \\ m\,T(n/m) & \text{if } n>m \end{cases}$$

T(n)=mT(n/m) ……(1)
If n=n/m
Replace it into ….(1)
T(n/2)=mT(n/m^2)
So, T(n)=m^2T(n/m^2)
.
.
T (n) $=m^{k\ T}(n/m^{k})$ where k=1,2,3..
Assume $n=m^{k}$
$T(n)=m^{k}\ T(m^{k}/m^{k})$
$T(n)=m^{k}\ T(1)$
$T(n)= m^{k}$ $n=m^{k}$
$\text{Log}_m n = \text{Log}_m m^{k} = k$

Complexity of the algorithm to make pairs to convert Binary to $\text{AHSD}_{(r)}$*:* ***O (log*** $_{m}$ ***n)***

2.4 Addition of AHSD $_{(4)}$ number system [1]

Here the addition process will be shown for AHSD number system. This addition process can be for 1-bit to n-bit adder design without considering the carry propagation and its delay as well.

Example: Here two 4-bit AHSD $_{(4)}$ numbers are added.

X→ 11 11 11 11
Y→ 11 00 01 00
……………
X (ahsd$_{(4)}$)→ (3 3 3 3)$_4$
Y (ahsd$_{(4)}$)→ (3 0 1 0)$_4$
…………….
X+Y(Z): 6 3 4 3

Transfer digit: 1 1 1 1 0 c
Interim sum: 2 -1 0 1 µ
……………………
Final sum S: (1 3 0 1 1)$_4$

Result=(01 11 00 01 01)$_2$=(453)$_{10}$

The final result is in binary format. The given example illustrates the addition process without carry propagation.

3 Proposed Design of adder using Neural Network

The neuron will work on the basis of feed-forward network with parallel processing. This technique can be viewed as doing addition in adder in parallel. The concept of parallel addition using neural network can be shown as the block diagram below.

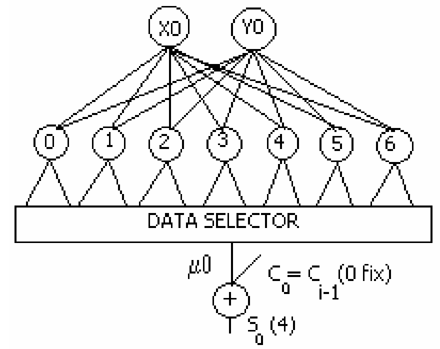


Figure 2: Trivial 1-bit AHSD radix-4 adder using NN.

The figure 2 shows the atomic view of the adder using Neural Network. If we make it more generalized then the figure will be just like the following.

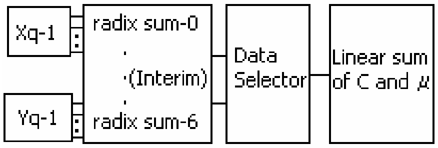


Figure 3: N-bit adder generalization.

The N-bit adder generalization is shown in figure 3.

Lemma: The total number of neurons for generating interim sum and carry of a radix-n asymmetric q-bit adder design is q×2(n-1).

Proof: As *n* is the radix, so each bit will contain the value of *n-1*. So the interim sum will be in the range of 0 to *2(n-1)*. The total number of neurons for 1-bit adder design will be *2(n-1)*. For the design of a q-bit adder, the total number of neurons will be *q×2(n-1)*.

Here we proposed an algorithm for n-bit adder from binary to AHSD $_{(4)}$ using NN.

Algorithm:

Step 1: Create 4^n input vectors (2n elements each) that represent all possible input combinations. For example, for 1-bit addition, there will be 4 input vectors (2 elements each): {0, 0}, {0, 1}, {1, 0}, {1, 1}.
Step 2: Create 4^n output vectors (ceiling of [$^n/_2$ + 1] elements) that represent the corresponding target combinations. For example, for 1-bit addition, there will be 4 output vectors (2 elements each): {0, 0}, {0, 1}, {0, 1}, {0, 2}.

Step 3: Create a feed-forward back propagation neural network.

Step 4: Train the neural network for the input and output vectors of *steps 1*and *step2*.

Step 5: Simulate the neural network with the input vectors of *step 1*, getting the sum in AHSD $_{(4)}$ as the target.

4 AHSD addition for Radix-5

The asymmetric high radix number system considering radix-5 is not well suited for addition. The first conversion from binary to AHSD requires pairing bits by 3-bit binary. It will convey the values from 0 to 7 in decimal. But radix-5 will need 0 to 4 values. So we will get 5, 6, 7 as undetermined values. Hence the addition process will not be possible as well. Thus radix-4 AHSD is best suited for addition that is carry free and fast.

5 Simulations

Here the few simulations are described. We performed the simulation using Matlab 7.1 neural network tools in a computer with Pentium IV processor. At first conversion from binary- to- AHSD numbers is done for the manual simulation, then to get self-consistent adjustment (SCA) and adjacent digit modification (ADM) two AHSD numbers is added. Then arithmetic addition of SCA and ADM is done. Now NN design for addition of AHSD is considered. Finally by using Mathlab-7.1 simulation tool simulation is performing for NN design and getting addition result of AHSD numbers. In the figure 4 this workflow procedure is shown.

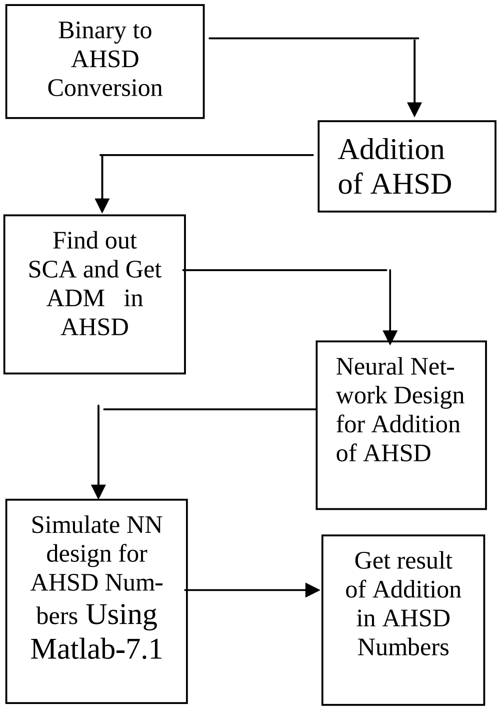

Binary to AHSD Conversion
Addition of AHSD
Find out SCA and Get ADM in AHSD
Neural Network Design for Addition of AHSD
Simulate NN design for AHSD Numbers Using Matlab-7.1
Get result of Addition in AHSD Numbers

Figure4: The Workflow of simulation procedure for AHSD numbers

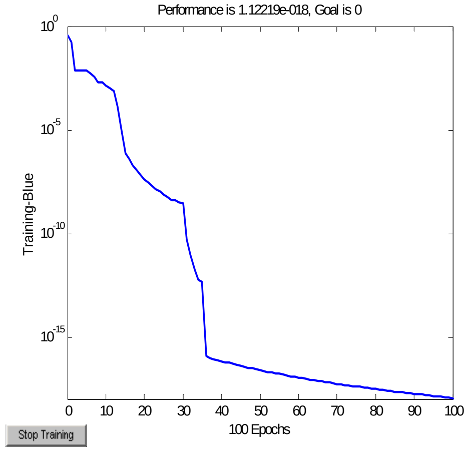


Figure 5 -1:(TF- Logsig) 1-bit adder simulation. and epoch is 100.

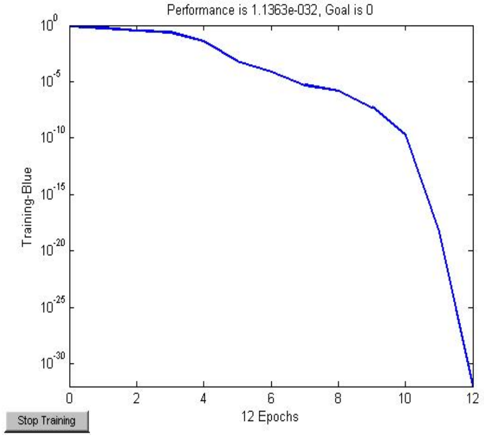


Figure 5-2: (TF-Tansig) 1-bit adder for simulation and epoch is 12.

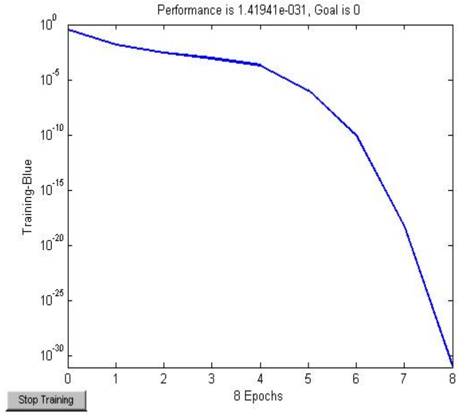


Figure 5-3: (TF-Purelin) 1-bit adder for simulation and epoch is 8.

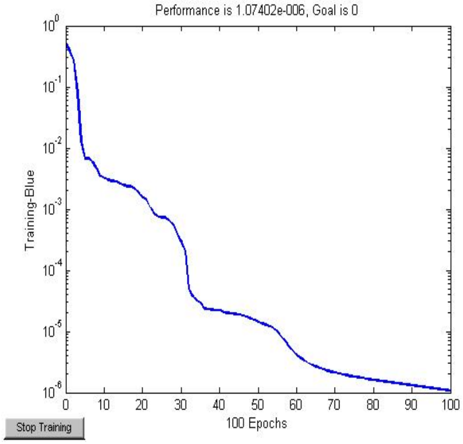


Figure 6-1: (TF-Tansig) 2-bit adder for simulation and epoch is 100 epochs.

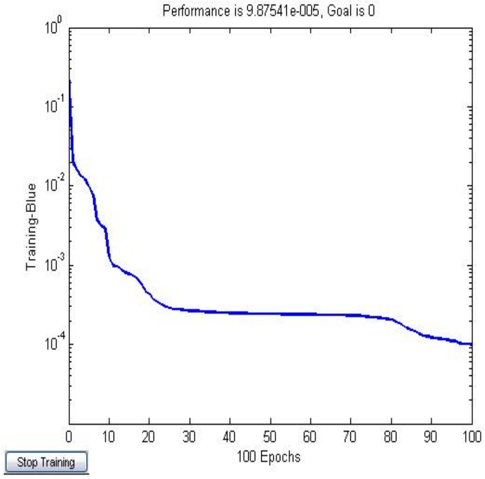


Figure 6-2: (TF-Logsig) 2-bit adder for simulation and epoch is 100 epochs.

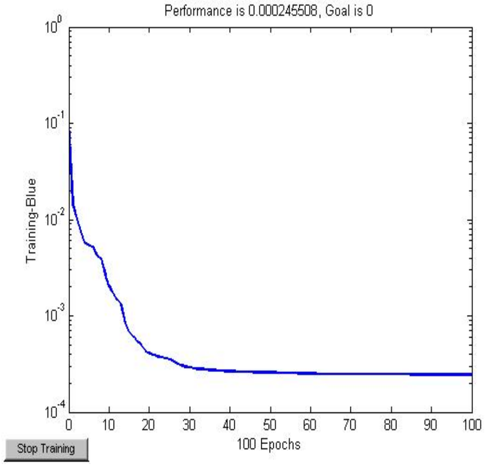


Figure 7: (TF-Logsig) 3-bit adder for simulation and epoch is 100 epochs.

6 Experimental Results

The table1 is summarized here from the experiments done.

Table 1: NN experimental results for AHSD adder design.

Bit	Layer	Neuron numbers	Activation Function
1-bit	2	I=2/O=2	Tansig/Logsig/Purelin
2-bit	2	I=4/O=2	Tansig/Logsig
3-bit	2	I=6/O=3	Logsig

Satisfactory performance is found in the design of AHSD-radix-4 adder using NN. This performance shows the better implementation of the adder of redundant number system than radix-2 signed digital adder using NN [5].

7 Conclusion

In this paper one CF digital adder for AHSD $_{(4)}$ number system based on NN has been proposed. Additionally, if $r = 2^m$ for any positive integer m, the interface between AHSD and the binary number system can be realized and it can be easily implement able. To make pairs at the time of conversion from binary to-AHSD, an algorithm has also been proposed. Besides, NN design with Matlab-7.1 simulation tool is used to achieve high speed for addition on AHSD. Even though hardware implementation is not done here but realization shows, the proposed NN design can be flexible. Since both the binary-to-AHSD and AHSD-to-binary converter .CF adder operate in a constant time, we can conclude that the AHSD-to-binary converter dominates the performance of the entire AHSD based on NN design system. The time complexity of the entire AHSD CF adder is O (log $_m$ n). Finally, it is appealed that additional studies is necessary including not only the implementation of CF adder with different design skills, but also the applications of various CF adders to design fast arithmetic. Hopefully, in future this paper can be extended for the digital adder of AHSD number system based on NN and which can be realized by using hardware, using current mode CMOS or voltage mode CMOS [6] and using SPICE simulation tool for hardware realization.

References

[1] S.H. Sheih, and C.W. Wu, " Asymmetric high-radix signed-digit number systems for carry-free addition," Journal of information science and engineering 19, 2003, pp.1015-1039.

[2] B. Parhami, "Generalized signed-digit number sys tems: a unifying framework for redundant number representations," IEEE Transac tions on Computers, vol. 39, 1990,pp.89- 98.

[3]S.H. Sheih, and C.W. Wu, "Carry-free adder design using asymmetric high- radix signed-digit number sys tem," in the Proceedings of 11th VLSI De sign/CAD Symposium, 2000, pp. 183- 186.

[4] M. Sakamoto, D. Hamano and M. Morisue, " A study of a radix-2 signed-digit al fuzzy processor using the logic oriented neural networks," IEEE International Systems Conference Proceedings, 1999, pp. 304-308.

[5] T. Kamio, H. Fujisaka, and M. Morisue, "Back propagation algorithm for logic oriented neural networks with quantized weights and multilevel threshold neurons," IEICE Trans. Fundamentals, vol. E84-A, no.3, 2001.

[6] A. Moushizuki, and T. Hanyu, " Low-power multi ple-valued current-mode logic using sub strate bias control," IEICE Trans. Electron., vol. E87-C, no. 4, pp. 582-588, 2004 .

Learning from Structured Data by Finger Printing

Thashmee Karunaratne
Department of Computer & Systems Sciences
Stockholm University and
Royal Institute of Technology
Forum 100
SE 164 40, Kista, Sweden
si-thk@dsv.su.se

Henrik Boström
Department of Computer & Systems Sciences
Stockholm University and
Royal Institute of Technology
Forum 100
SE 164 40, Kista, Sweden
henke@dsv.su.se

Abstract

Current methods for learning from structured data are limited w.r.t. handling large or isolated substructures and also impose constraints on search depth and induced structure length. An approach to learning from structured data using a graph based canonical representation method of structures, called finger printing, is introduced that addresses the limitations of current methods. The method is implemented in a system, called DIFFER, which is demonstrated to compare favourable to existing state-of-art methods on some benchmark data sets. It is shown that further improvements can be obtained by combining the features generated by finger printing with features generated by previous methods.

1 Introduction

In many domains, in which a model is to be generated by machine learning, examples are more naturally represented by structured terms than fixed-length feature vectors. For example, in chemo informatics, molecules are naturally represented as two or three dimensional structures of atoms. Another example is when having data on XML format, which could be directly mapped on tree structures.

Several approaches to learning to classify structured data have been introduced in the field of machine learning. The structure classification problem has been addressed as a rule learning problem [1,2,3,4,5], as a graph mining problem [6,7,8,9,10,11] and as a propositionalization problem [12,13,14]. The methods address two main varieties of problems. The first category is discovery methods of features that best discriminate between the classes. Krogel et al [12], for example, introduce an approach to selecting "most interesting" features of structured data that discriminate between the classes. A key requirement of the feature discovery methods is that the discovered features should be comprehensible.

In contrast to discovery methods, classification methods generate global models for classifying all examples, but the models need not necessarily be comprehensible. Most classification methods assume that all examples can be represented by fixed-length feature vectors, and finding features that suitably contain the relevant information in this format, can be considered a major knowledge engineering bottleneck for these types of method. This is true in particular when the examples are most naturally represented as structured terms (e.g., trees, lists, etc.). Existing methods for structure classification are limited w.r.t. the complexity and volume of structured data, which is described in detail in section 2, and hence more robust methods for learning from structured data are needed. The method presented in this paper, which extracts features from structures by a method called *finger printing,* is motivated exactly by this need.

The rest of the paper is organized as follows. Section 2 discusses the state-of-art structure classification methods and their limitations. Section 3 introduces the novel finger printing method, which implementation is described in section 4. Section 5 presents an empirical evaluation, comparing the novel method to some state-of-the-art methods on some benchmark datasets. Finally, section 6 describes the concluding remarks and possible further extensions to the demonstrated theory.

2. Current Approaches to Learning from Structured Data

Current state-of-art methods for feature discovery and classification use several forms of structure transformation. Inductive logic programming [5] has drawn immense popularity since its inception, mainly due to that background knowledge and data as well as the result of the methods are represented in the same format: logic programs. Propositionalization methods is one class of ILP methods that transform the relational rule learning problem into a standard attribute-value learning problem by identifying suitable features [12,15]. However, these, as well as the standard ILP methods, are often faced with a huge search space, either for which constraints have to be imposed, or the domain has to be restricted in terms of the number of examples considered [15]. The limits on search depth and clause length typically result in that the substructures discovered by ILP methods are quite small and usually are limited to 5-6 structural relations [16].

Graph mining methods are efficient enough to discover considerably large substructures, unlike the propositionalization methods [9,16]. Although the current algorithms already perform quite well, they still have some limitations. One of these is that the discovered graphs are by necessity connected. This prevents inclusion of isolated or far away frequent nodes or sub graphs. Thus two fragments within a graph that are not connected are not being considered in conjunction by current methods, even if the contribution of these fragments when taken together would be a highly potential feature. Another limitation of current graph mining methods is that they only consider exact matches of the sub-graphs and hence do not allow mining "similar sub-graphs" [17], i.e., sub-graphs that are not exactly equal to each other, but differ only by a few nodes. For example, in a chemoinformatics application, different molecules may have carbon chains of different lengths, but other atoms such as nitrogen and oxygen are connected to the carbon chains in a same topology, i.e., substructures differ only by its length of the carbon chain. These substructures are not equal to each other since they differ by the length of the carbon chain, but rather "similar" since the topology of the substructure is same. Current methods consider these substructures as completely different, since the substructures do not exactly match with each other. A further description about similar subgraphs can be found in [17]. Furthermore, memory and runtime are challenges for most of the graph mining algorithms [17]. It is an open question how to realize the class distribution over sub-graphs without searching different branches of the lattice several times [7]. The need for inexact graph matching might become more and more important in this context. As Washio & Motada [7] reports "Even from a theoretical perspective, many open questions on the graph characteristics and the isomorphism complexity remain. This research field provides many attractive topics in both theory and application, and is expected to be one of the key fields in data mining research".

In summary, ILP/propositionalization methods can be useful for learning from structured data if discovery of small substructures is sufficient, but they do require that non-trivial constraints on the search space are provided. If the domain of interest requires the discovery of large substructures, graph mining methods are often more suited. However, these cannot be used to discover several isolated substructures and require exact matching of substructures. Hence, there are demands for much robust methods for learning from structured data.

3. Finger Printing

Our approach to structure classification employs a graph transformation method which could address some of the limitations discussed in the previous section. The method has the ability to combine isolated substructures and has the potential to discover "similar substructures". It also does not require any constraint to be imposed on the search space. Our method follows a data to model (bottom – up) search strategy and digs down any potential substructures irrespective of its length. Since the graphs are transformed into a canonical form called *finger print*, which is a hashed form of a sparse vector of zeros and ones, the computational cost in manipulation of the graphs are considerably low. Our method could be applied to any form of structured data, from trees to undirected graphs, from sequences to tuples etc., and hence all these types of structured data are referred to as graphs during the rest of this paper.

3.1 The finger printing method

Several methods have been suggested to represent structured data for learning algorithms, and canonical forms of graphs are among the most popular due to their computational simplicity. Our method of transforming graphs into a canonical form is called *finger printing*. The method first includes a preprocessing step in which each structured data is represented by a labeled graph in the *forest*, and then transforming each graph into an adjacency matrix using the definitions 1 and 2 given below.

Definition 1 *Foresting: Let G be the set of graphs. Then the forest $F(V_F,E_F)$ of G is denoted as $\bigcup_{i=1}^{n} g_i$, the union of g_i where n is the number of elements in G. Further V_F is $\bigcup_{i=1}^{n} V_i$ and, E_F is $\bigcup_{i=1}^{n} E_i$.*

An example of a forest *F* of a set of graphs *G* is given in Figure 1.

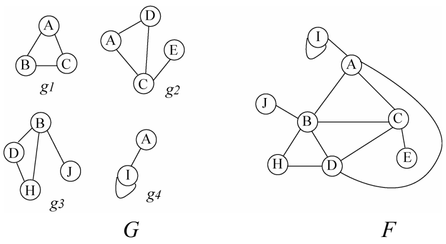


Figure 1: The forest *F* of *G*

Definition 2 *Adjacency matrix of a graph g of a forest F: The adjacency matrix of a graph g is an n x n matrix $A = (a_{i,j})$ in which the entry $a_{i,j} = 1$ if there is an edge between vertex i and vertex j in g and $a_{i,j} = 0$ if there is no edge between vertex i and vertex j in g, where n is the number of vertices in forest F.*

We construct the finger print of a graph *g* and an associated forest *F* using definition 4 given below. In doing so the adjacency matrices of each graph is transformed into a vector *v* of (1 × *n***n*), where *n* is the dimension of the adjacency matrix. The vector *v* has the characteristic of being a sparse vector.

Definition 3 *Sparse vector v: Let $g \in G$ and A_g be the adjacency matrix of g. The sparse vector v of g is a vector of ones and zeros of dimension ($1 \times n*n$), such that* $v = \bigcup_{i=1}^{n}[\bigcup_{j=1}^{n} a_{ij}]$ *where i is the row number and j is the column number of the adjacency matrix.*

Definition 4 *Finger print: A finger print is a mapping $f: x \rightarrow y$, such that if $x=y$ then $f(x) = f(y)$ and if $f(x) = f(y)$ then $x=y$, where x and y are two sets of consecutive characters.*

Accordingly, for the example in Figure 1, *n* = 8 and the sparse vectors of the graphs in Figure 1 would be of length 64. Hence *v*(g1) = (01100000101000001100000000000000000000000 00000000000000000000000). Under the similar transformation rule, the sparse vector of g2, is (00110000000000001001100010100000001000000 00000000000000000000000). The sparse vectors are hashed with respect to the position of presence of ones in each sparse vector in order to reduce the storage space. The hashed sparse vector of *v*(g1) is (2,3,9,11,17,18), and the hashed sparse vector of *v*(g2) is (3,4,17,20,21,25,27,35). According to the Definition 4 this vector possesses the property of a finger print *f.* Hence *f*(g1) = (2,3,9,11,17,18) and *f*(g2) = (3,4,17,20,21,25,27,35).

3.2 Substructure search using finger prints

Each position in the finger print represents a connection between two nodes of the graph g, and hence is a substructure of the graph. For example the first digit of *f*(g1), i.e., 2, represents the edge A-B between the nodes A and B in the graph g1. This segment of the graph g1 is a subgraph, and hence represents a substructure in the respective structure. If we expand this notation to two graphs, we could find common elements of two graphs g1 and g2, which represent substructures common to both the graphs under consideration. Taking the same example from Figure 1, the common elements of *f*(g1) and *f*(g2) is the intersection of the two finger prints. i.e., $f(g1) \cap f(g2)$ = (3,17). We use this property in searching for commonality between finger prints of the graphs in order to find the most discriminating features.

During the search of common substructures, a pair-wise substructure search algorithm is used to filter the maximal common substructures. This algorithm is "pair-wise" since it compares two finger prints at a time, and the searched substructure is "maximal" since we keep the largest common subset of numbers in the two finger prints under consideration. This substructure search algorithm, given in Figure 2 below, finds all possible pair-wise maximal substructures within the training examples.

```
function PairWiseMaximalSubstructures((f1, ....., fn ); list
of finger prints)
  j=1
 while (j ≤ n – 1 )
do
    k = j+1
    while (k≤ n)
      do
       s[j,k]= s[k,j]=fj ∩ fk
      add s[j,k]and s[k,j] to MaximalSubstructures
      k++
   done
  j++
done
return MaximalSubstructures
```

Figure 2: The pair-wise maximal substructure search algorithm

3.3 Feature construction

The feature set used for classification is the most discriminative set of substructures found by the pairwise maximal substructure search algorithm. We use the standard covering statistic for finding the most discriminating set of substructures.

Definition 5 *Covering Statistic: The coverage of a substructure s_i is defined as*

$$C(s_i) = p(s_i) = n\,(s_i)/\,N$$

where n (s_i) is the number of examples for which the substructure is present in the example, and N is the total number of examples present in the training set.

Each substructure discovered using the maximal common substructure search algorithm is evaluated by the coverage statistic given in Definition 5. The weighted substructure set is then ranked in descending order. We also use a maximum and a minimum threshold in order to filter the most discriminating set of substructures. This set of substructures obtained through the search and filter procedure are the most discriminating feature set for the domain of examples. This feature set is then used when building the classification model.

4. Implementation

We have developed a feature construction and classifier system called DIFFER (**DI**scovery of **F**eatures using **F**ing**ER** prints), using the methodology described in section 3. DIFFER handles examples containing structured data. Therefore inputs to the system are the structures. DIFFER produces an output containing the derived feature set and the presence/non-presence of those features in examples in a form of a text file that can be used by most standard classification methods (at the moment the output file is of the format of .arff which is the recognizable format for WEKA data mining toolkit). In summary the main features of DIFFER are:

- any type of structured data can be handled such as trees, graphs, tuples, strings etc. etc.
- the canonical form used for graph transformation is the finger print
- isolated sub-graphs/nodes can be identified. i.e, the substructures need not to be necessarily connected.
- similar sub-graphs can be identified as well. i.e., the substructures treated as common are not equivalent to each other but "similar".
- There are no constraints on length of the searched substructure, and therefore subgraphs of any size may be identified.

5. Experimental Evaluation

We have used some benchmark datasets to compare the performance of DIFFER with other available methods for learning from structures. One benchmark dataset concerns predicting mutagenicity on *Salmonella typhimurium*, which comprises of 230 molecules. Debnath et. al. [18] grouped 188 examples out of these as "regression friendly". This subset contains 125 examples with a positive log mutagenesis whereas the remaining 63 examples are inactive with respective to log mutagenesis. We have used this regression friendly subset of the mutagenesis dataset for our experimental evaluation. This is a two class problem of predicting whether a compound is mutagenic or not.

The second benchmark data set concerns the very popular east-west train problem [19], which contains 20 trains where 10 each are headed to east and west respectively. The task is to identify the characteristics of the trains that make them headed east or west. The third dataset, carcinogenesis is also, like the first, from the domain of chemo-informatics. The dataset was originally developed within the US national toxicology program [20]. It consists of 298 compounds that have been shown to be carcinogenic or not in rodents. Although the original dataset contains 3 classes of carcinogenesis, these were treated as one class as done in most previous studies.

The three benchmark datasets were given as input to DIFFER and a summary of the outcome of the method is given in Table 1 below.

Dataset	No. of examples	features selected[1]
Muta	188	171
Trains	20	30
Carci	298	154

Table 1: Summary of induced features for benchmark datasets

We have used all the data as training examples during feature generation. This does not impose any bias on feature construction since we are not considering class distribution of features during the feature construction. Transformation of structures into graphs is carried out according to the definition 2.

[1] Since the set of features discovered by Maximal common substructure search algorithm is small we did not apply any threshold here.

For the first and third datasets, molecules are represented by graphs in such a way that a node of graph is an atom in the molecule with its label (carbon for example), plus the bonds attached to the atom. For example, a carbon atom with two single bonds and a double bond is converted to a labeled node c[112]. The second dataset contains trains as its structures, and each train has cars with are described by a set of properties, e.g., shape, no. of wheels etc. A node in this domain is represented by a tuple *car(<properties>)*. For example a car with a long rectangular shape, a flat roof, sides that are not double and 3 wheels, is represented by *car(rectangle,long,not_double,flat,3)*.

Features generated by DIFFER is used as input to a standard machine learning method. The method used in this experiment is random forest [21] with 50 trees and where 10 random features are evaluated at each node, as implemented in the WEKA data mining toolkit [22]. 10 fold cross validation is used as the evaluation method. The results we obtained with DIFFER were compared with existing state-of-the-art methods, including a propositionalization method, RSD [12] and two graph mining methods, SUBDUE-CL [9] and Tree$^2\chi^2$ [11]. We have used all the data in each of the 3 benchmark datasets as training examples for RSD as well. The WEKA [22] implementation of random forest of 50 trees with 10 random features for evaluation at each node is used for reproduction of accuracies in RSD. 10 fold cross validation is used as the evaluation method. We did not reproduce the results for SUBDUE-CL and Tree$^2\chi^2$, but give the accuracies reported in [9] and [11] respectively. The results are summarized in Table 2.

Dataset	Accuracy				
	DIFFER	RSD	SUBDUE-CL	Tree$^2\chi^2$	DIFFER + RSD
Trains	80%	75%	-	-	85%
Mutagenesis	80.61 %	88.86 %	-	80.26%	92.76 %
Carcinogenesis	65.25 %	54.37 %	61.54 %	-	65.33 %

Table 2: Comparison of DIFFER with some state-of-the-art methods

These results show that DIFFER may perform as well as or better than existing methods. We also have studied what happens when merging features of DIFFER with those of other methods. When merging the feature set of RSD with the feature set of DIFFER, an increase in accuracy was observed (final column of Table 2). We analyzed the feature set generated by RSD for the mutagenesis dataset and rather surprisingly, we found that it did not contain any atom-bond features. Nonetheless it contained global molecular structure properties such as whether or not two connected nitro groups are present. In contrast to this, the features generated by DIFFER contains inner structural information of atom-bond connections. The experiment demonstrates that by merging these two complementary sets of features the accuracy of the resulting model can be increased.

6. Concluding Remarks

Learning from structured data is an important challenge for machine learning methods, with many important applications, for example within analyzing data from the web, in chemo- and bioinformatics, in management and business transaction domains. These domains are often complex not only in terms of the presence of structures, but also often in terms of the size of the data sets to be analyzed. Existing techniques for learning from structured data are demonstrated to have a number of limitations w.r.t. to effectively analyzing the data due to inability to discover isolated sub-graphs or capture topology of similar sub-graphs and by requiring that non-trivial constraints on the search space is provided, something which may prevent the discovery of large interesting substructures. In order to overcome these limitations, a novel method, that transforms structured data into a canonical representation, called finger prints, has been presented.

The new method, which has been implemented in a system, called DIFFER, has been shown to be competitive with the existing state-of-the-art methods on some standard benchmark data sets, without imposing constraints on the search space. The reason for its effectiveness can be explained by its ability to mine large as well as isolated discriminative sub-graphs. A very interesting observation is that the classification performance can be improved by merging the features generated by DIFFER with features generated by other methods and thereby integrating the different qualities of several methods. Thus rather than searching for new feature extraction methods that on its own compete with existing methods, it appears to be a promising approach to search for new methods that generate complementary features.

There are several possible directions for future work. At present DIFFER's substructure search is a pair-wise approach, for which the computational cost grows quadratically with the number of examples. A more efficient procedure could be obtained by using some incremental way of searching for the substructures. Sampling of which pairs to consider is also a straightforward way of controlling the computational cost [3]. Alternatives to the use of the covering statistic in conjunction with maximum and minimum thresholds could also be explored. Candidates for this include model driven approaches such as voting by the convex hull or a coverage measure.

The promising result of combining the features generated by DIFFER and RSD also leads to considering merging the features of DIFFER and other methods, perhaps further improving the predictive performance.

References

[1]. Zaki, M.J., Aggarwal, C.C. (2003), "*XRules: an effective structural classifier for XML data*" KDD, Washington, DC, USA, ACM 316–325

[2]. Quinlan, J. R., Cameron-Jones, R. M., (1993), "*FOIL"*, Proceedings of the 6th European Conference on Machine Learning, Lecture Notes in Artificial Intelligence, Vol. 667, pp. 3-20. Springer-Verlag (1993)

[3]. Muggleton, S.and Feng. C., (1992), "*Efficient induction in logic programs*", Inductive Logic Programming, pages 281-298. Academic Press

[4]. Srinivasan A,. King, R.D, and Muggleton S, (1999), "*The role of background knowledge: using a problem from chemistry to examine the performance of an ILP program*", Technical Report PRG-TR-08-99, Oxford University Computing Laboratory, Oxford, 1999.

[5]. Muggleton, S., (1995), "*Inverse entailment and Progol*", New Generation Computing, Special issue on Inductive Logic Programming, 13(3-4):245-286

[6]. Cook, J. and Holder, L., (1994), "*Substructure discovery using minimum description length and background knowledge*", Journal of Artificial Intelligence Research, 1:231-255

[7]. Washio T. and Motoda, H. (2003), "*State of the Art of Graph-based Data Mining*", SIGKDD Explorations Special Issue on Multi-Relational Data Mining, pp 59-68, Volume 5, Issue 1

[8]. Dehaspe, L. and Toivonen, H, (1999). "*Discovery of frequent datalog patterns*", Data Mining and Knowledge Discovery, 3(1):7-36

[9]. Gonzalez, J., Holder, L. B. and Cook, D. J. (2001), "*Application of Graph-Based Concept Learning to the Predictive Toxicology Domain",* Proceedings of the Predictive Toxicology Challenge Workshop

[10]. De Raedt, L. and Kramer, S., (2001), "*The levelwise version space algorithm and its application to molecular fragment finding*" In IJCAI'01: Seventeenth International Joint Conference on Artificial Intelligence, volume 2, pages 853-859

[11]. Bringmann, B., and Zimmermann, A., (2005), "*Tree - Decision Trees for Tree Structured Data*", Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2005), Notes in Artificial Intelligence, (LNAI) 3721, pp. 46-58, Springer (2005)

[12]. Krogel, M-A., Rawles, S., Železný, F., Flach, P. A., Lavrač, N., and Wrobel, S., (2003). "*Comparative evaluation of approaches to propositionalization*", Proceedings of the 13th International Conference on Inductive Logic Programming (ILP'2003), number 2835 in Lecture Notes in Computer Science, pages 197--214, Springer Verlag.

[13]. Lavrac, N. and Flach P., (2000), "*An extended transformation approach to Inductive Logic Programming*", University publication, Department of Computer science, University of Bristol

[14]. Lavrac N., Zelezny F., Flach P., (2002): "*RSD: Relational Subgroup Discovery through First-order Feature Construction*", Proceedings of the 12th International Conference on Inductive Logic Programming (ILP'02), Springer-Verlag, ISBN 3-540-00567-6

[15]. Nattee, C., Sinthupinyo, S., Numao, M., Okada, T., (2005), "*Inductive Logic Programming for Structure-Activity Relationship Studies on Large Scale Data*", SAINT Workshops 2005: 332-335

[16]. Inokuchi, A., Washio, T., and Motoda, H., (2003), "*Complete mining of frequent patterns from graphs*" Mining graph data, Machine Learning, 50:321-354

[17]. Fischer, I. and Meinl, T., (2004), "*Graph based molecular data mining - an overview*", IEEE SMC 2004 Conference Proceedings, pages 4578--4582

[18]. Debnath, A.K. Lopez de Compadre, R.L., Debnath, G., Shusterman, A.J., and Hansch, C. (1991), "Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds: Correlation with molecular orbital energies and hydrophobicity", Journal Med. Chem. 34:786-797

[19]. Michie,D., Muggleton,S., Page,D., and Srinivasan,A., (1994), "*To the international computing community: A new East-West challenge*" Oxford University Computing laboratory, Oxford, UK, URL:

ftp://ftp.comlab.ox.ac.uk/pub/Packages/ILP/trains.tar.Z

[20]. US National Toxicology program, http://ntp.niehs.nih.gov/index.cfm?objectid=32BA9724-F1F6-975E-7FCE50709CB4C932

[21]. Breiman, L., (2001), "*Random Forests*", Machine Learning 45(1): 5-32 (2001)

[22]. Ian H. Witten and Eibe Frank (2005) "Data Mining: Practical machine learning tools and techniques", 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

Haptic Perception with a Robotic Hand

Magnus Johnsson[⋆]
[⋆]Dept. of Computer Science and
Lund University Cognitive Science
Lund University, Sweden
Magnus.Johnsson@lucs.lu.se

Christian Balkenius[†]
[†]Lund University Cognitive Science
Lund University, Sweden
Christian.Balkenius@lucs.lu.se

Abstract

We have developed an 8 d.o.f. robot hand which has been tested with three computational models of haptic perception. Two of the models are based on the tensor product of different proprioceptive and sensory signals and a self-organizing map (SOM), and one uses a novel self-organizing neural network, the T-MPSOM. The computational models have been trained and tested with a set of objects consisting of hard spheres, blocks and cylinders. The first computational model, which is based on the tensor product, was able to discriminate between all the test objects. The second and third models could also discriminate between the test objects, but in addition they were capable of shape categorization.

1 Introduction

Haptic perception is an active tactile process that involves both sensory and motor systems to identify an object. In the human hand, somatosensory information is provided by receptors in the skin, the muscles and the joints. This information is needed to calculate the path of the hand, to control its locomotion and to enable an adequate grasp.

To some extent, objects can be identified with passive touch. In this case, the identification is based on the excitation of specific receptors sensitive to pressure, heat, touch and pain. The combination of these sensations provides information for object identification (Millar, 2006). A combination of cutaneous and proprioceptive information about the configuration of the hand is needed for better discrimination.

Surprisingly little work has addressed the problem of haptic perception. Most models of hand control have focused on the motor aspect rather than on haptic perception (Arbib, Billard, Iacoboni & Oztop, 2000; Fagg & Arbib, 1998). This is also true for the robotic hand research which has mainly looked at grasping and object manipulation (DeLaurentis & Mavroidis, 2000; Sugiuchi, Hasegawa, Watanabe & Nomoto, 2000; Dario, Laschi, Menciassi, Guglielmelli, Carrozza & Micera, 2003; Rhee, Chung, Kim, Shim & Lee, 2004). There are some exceptions however. Dario et al (2000) developed a system capable of haptic object classification. Another example is a system made for studies of the interaction between vision and haptics (Coelho, Piater & Grupen, 2001). Hosoda, Tada and Asada (2006) have built an anthropomorphic robot finger with a soft fingertip with randomly distributed embedded receptors. Jockusch, Walter and Ritter (1997) have developed a cost effective artificial fingertip with force/position sensors and slippage detection.

Heidemann and Schöpfer (2004) have developed a system for haptic object identification that uses a low-cost 2D pressure sensor with coarse resolution. The sensor is mounted on a PUMA-200 industrial robot arm. The system collects information by repeated contacts with the object. The collected information is combined is used as input to a three-step processing architecture that lets features form automatically.

Our previous research in haptics consists of the design and implementation of a simple three-fingered robot hand, the Lucs Haptic Hand I, together with a series of computational models (Johnsson, 2004, 2005; Johnsson et al., 2005a, 2005b; Johnsson & Balkenius, 2006a). This paper describes its successor, the Lucs Haptic Hand II, and three haptic models developed for it. The Lucs Haptic Hand II (Fig. 1) is an 8 d.o.f. three-fingered robot hand equipped with 45 pressure sensors developed at Lund University Cognitive Science (Johnsson & Balkenius, 2006b). Each finger consists of a proximal finger segment articulated against a triangular plastic plate, and a distal finger segment articulated against the proximal segment. Each finger segment contains a RC servos. A sensor plate is mounted on the inner side of each finger segment and contains 7 or 8 pressure sensitive sensors.

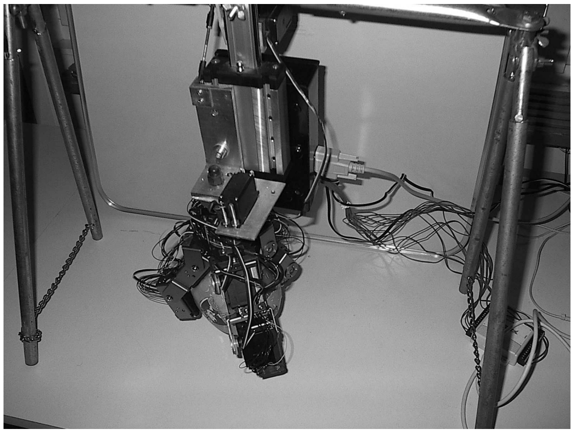


Figure 1: *The Lucs haptic hand II while grasping a soft ball. The 8-dof robot hand has three fingers, each consisting of two segments symmetrically mounted on a triangular plastic plate. The hand is equipped with a wrist and a lifting mechanism. The finger segments are built with RC servos mounted with servo brackets. All the actuators of the Lucs Haptic Hand II are controlled via a SSC-32 (Lynxmotion, Inc.). At each finger segment there is a plate mounted. The plates are equipped with push sensors.*

The triangular plastic plate is mounted on a wrist consisting of a bearing, and an actuator connected to a rod for force transmission. The wrist enables horizontal rotation of the robot hand. The wrist is in turn mounted on a lifting mechanism consisting of an actuator and a splint. Fig. 2 depicts the sensors status during a moment in a grasping movement. A movie showing the Lucs haptic hand II in a grasping task is available on the web site (Johnsson, 2005).

Model 1 uses the tensor product (outer product) to combine cutaneous and proprioceptive information gathered by the robot hand. The tensor product is an operation between a n-dimensional column vector $x = (x_1, \ldots, x_n)^T$ and a m-dimensional row vector $y = (y_1, \ldots, y_m)$ resulting in a $n \times m$ matrix M, where

$$\mathbf{M} = \begin{pmatrix} x_1y_1 & x_1y_2 & \cdots \\ x_2y_1 & x_2y_2 & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

Model 2 uses the tensor product to combine the tactile information in several steps. Model 3 is similar to Model 2 but substitutes the tensor product operations with Tensor Multiple Peak Self-Organizing Maps (T-MPSOM:s), a novel neural network architecture that combines the computations of the tensor product with the merits of self-organizing maps. The aim of these models was to take another step in our research and implement a system capable of haptic shape categorization.

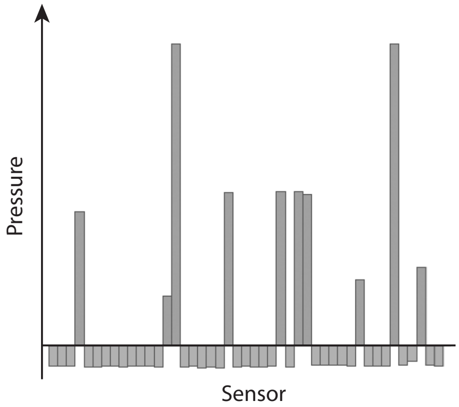


Figure 2: *Signals from the Lucs Haptic Hand II during the grasping of an object.*

2 Model 1

2.1 Design

The software for the Lucs haptic hand II is developed as Ikaros modules (Balkenius & Morén, 2003).

Beside the Lucs Haptic Hand II and the sensory and motor drivers, model 1 (Fig. 3) consists of four common Ikaros modules.

The Grasping Module takes care of the individual grasping movements, i.e. not the changes in the height of the robot hand or the angle of the wrist. When executing a grasping movement the module starts by moving the proximal finger segments. The sensor status is measured individually for each finger segment and a finger segment stops its movement when the total change of sensors registering exceeds a threshold or the position of the finger segment has reached a maximal allowed position. When the proximal segment of a finger stops, the distal segment starts to move. It continues to move until the total change of sensors registration for that particular segment exceeds a certain threshold or the position of the segment reaches a maximal position. The idea here is to let the robot hand take a shape that is in accordance with the grasped object. The Grasping Module controls the motor driver and also receives input from it. This input is a representation of the current configuration of the robot hand and can be thought of as proprioceptive information. The proprioceptive information is coded as a vector. This vector can be considered as a series of sequences of 10 elements, where each sequence code for the position of a moveable part of the Lucs Haptic Hand II. The position of a moveable part is coded by setting one of the elements in the sequence to 1 and the other elements to 0. In the second and the third models this vector is split into three, one that codes for the wrist angle, one that code for the height of the robot hand and one that code for the finger segments. The Grasping Module also receives information about the sensors status as input from the sensory driver. This tactile information is coded by a vector with 45 elements corresponding to the 45 sensors. Upwards in the model, the Grasping Module communicates with the Commander Module and with the STM Module. The input from the motor driver is combined with the input from the sensory driver by the use of tensor product between the proprioceptive vector and the sensory vector. This is calculated in the Grasping Module and is sent as output to the STM Module.

The Commander Module is responsible for the exploration of the object. This is done by carrying out a sequence of nine different grasps at two different heights and with 5 different wrist angles. In the model the haptic exploration is implemented by letting the Grasping Module receive orders about individual grasping movements and at what height and wrist angle the individual grasping movement is going to take place. The exploration continues until a sequence of nine exploration grasps have been executed with the actual object. The Commander Module receives signals from the Grasping Module about the status of the grasping movements, i.e. whether it is in progress or completed.

The STM (Short-Term Memory) Module receives the tensor product matrix from the Grasping Module, and the matrices from the whole exploration of an object are superimposed in the STM Module. Therefore the tactile information from the beginning to the end of the exploration of an object are put together and represented with one matrix. When haptically exploring an object the sensory information should, in raw or in refined form, be stored temporarily in the brain before the recognition of the object happens after some active exploration.

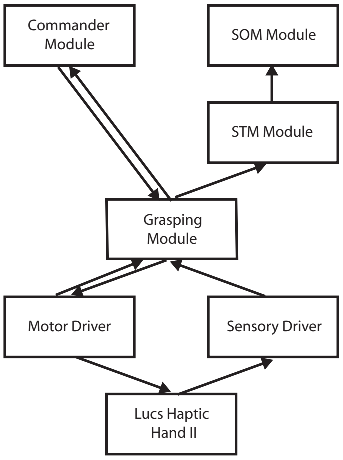


Figure 3: *Schematic depiction of model 1. The Lucs Haptic Hand II conveys the status of the sensors to the Sensory Driver which in turn conveys the status of the sensors as an array to the Grasping Module. The Motor Driver conveys, to the Lucs Haptic Hand II, the wanted positions of the servos in the robot hand and the wanted time to reach these positions. In addition, the Motor Driver conveys proprioceptive information, i.e. information about the configuration of the robot hand, the wrist and the lifting mechanism to the Grasping Module. The Grasping Module conveys the wanted configuration of the robot hand and the wanted time to reach it to the Motor Driver, the status of individual grasps to the Commander Module, and the Tensor product between proprioceptive and sensory information to the STM Module. The STM Module conveys a matrix of the tensor product superimposed from a whole exploration to the SOM Module. The Commander Module conveys orders to initiate grasping movements to the Grasping Module.*

The SOM Module When the exploration of the object is completed the output matrix of the STM Module that represents all tactile information from the exploration is used as input to the SOM Module. The SOM Module is a self-organizing neural network (Kohonen, 2001) with 225 neurons. In this module the learning and the categorization of this model takes place.

2.2 Grasping Tests

In order to test the model, we have used 6 different test objects, 2 different blocks, 2 different cylinders and 2 different spheres, see Table 1. To avoid a big influence of the hardness of the objects on the results, all the objects were made of hard materials like wood, hard plastic, hard board and metal.

To simplify the test procedure the system explored each of the 6 different objects 5 times, i.e. in total 30 explorations, and the output matrices from the STM Module were written to files. This set of 30 files was then used as a training and test set for the SOM Module. The model was trained with 1000 randomly chosen samples from the training set, then the trained model were tested with all 30 samples in the training set.

2.3 Results and Discussion

The result from the grasping test with this model is depicted in Fig. 4. In the figure the center of activity for each exploration sample in the training set is depicted. As can be seen the objects are categorized in a reproducible way. Also observe in the figure that the blocks in the lower part of the figure can be separated from other objects, i.e. from the spheres and the cylinders. An interpretation of this is that objects (in the training set) that are radial symmetric in at least one plane (the spheres and the cylinders) are separated from objects (in the training set) that are not radial symmetric (the blocks). Taken together, this means that the model is capable of learning to distinguish between individual objects and, at least to some extent, of learning to distinguish between shapes. Another observation that can be done is that many centers of activity are equal for different explorations of the same objects. We were very exact when localizing the objects for the hand to explore, i.e. an object was placed in exactly the same way in different explorations. When we carried out another test in which we were not that careful when locating the object in exactly the same way it did not work out that well. The conclusion from this was that the differences between the matrices generated in the STM Module were extremely small and a very exact location of the objects was needed in order for it to work. This also explains why the center of activation in the SOM Module was often the same in different explorations of the same object.

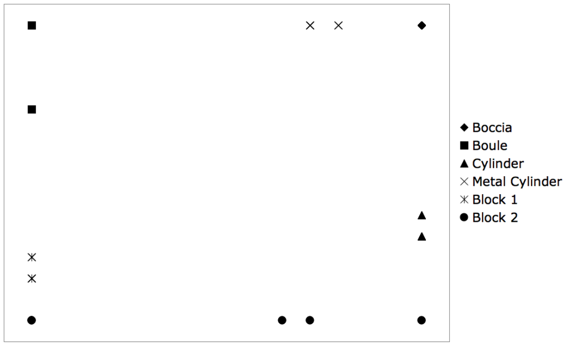


Figure 4: *Results of the grasping experiment with model 1. The depiction shows the centers of activity in the SOM during the testing with the training set patterns.*

3 Model 2

3.1 Design

To overcome the limitation of model 1 which needed the test objects to be accurately localized, we redesigned it into model 2, which uses the tensor product in three steps. Model 2 consists of the Lucs Haptic Hand II, the sensory and motor drivers, the four Ikaros modules used in model 1, and in addition three instances of an Ikaros module dedicated to carrying out the tensor product operation. In model 2 the Grasping Module does not receive input from the motor driver and does not calculate the tensor product. Instead the tensor product calculations are done in the dedicated Ikaros module.

In this model the proprioceptive information is divided into three vectors. One vector represents proprioceptive information about the height of the hand, one vector represents proprioceptive information about the angle of the wrist, and finally one vector represents the configuration of the hand.

One instance of the Tensor Product Module takes as input the part of the proprioceptive output vector from the motor driver that represents the height of the hand and the part that represents the angle of the wrist. The output from this instance of the Tensor Product Module is sent to another instance of the same module. In addition, the second instance of the module takes the vector that represents the configuration of the hand as input. The output from the second instance of the Tensor Product Module is conveyed to a third instance of the same module that also takes the sensory vector from the sensory driver as input. The result of this chain of operations is a successive recoding of the spatial coordinate of the sensor at each joint which makes the final matrix implicitly code the three dimensional location of the sensor when it reacts to the object. This is an important difference from model 1 where different shapes in principle could result in the same coding.

The resulting matrix is sent as output to the STM

Table 1: *The test objects used with the three haptic models for the LUCS Haptic Hand II*

Object	Material	Size (mm)	Size (mm)	Size (mm)
Boccia	Plastic	Diameter = 72	-	-
Boule	Metal	Diameter = 82	-	-
Cylinder	Hard Board	Diameter = 62	Height = 121	-
Metal Cylinder	Metal	Diameter = 75	Height = 109	-
Block 1	Wood	Height = 110	Length = 50	Width = 50
Block 2	Wood	Height = 110	Length = 58	Width = 50

Module. This module integrates the all the sensor readings over the object and forms a code that depends on the three-dimensional shape of the object. As in model 1, the output from the STM Module is used as input to a SOM Module with 225 neurons.

3.2 Grasping Tests

The grasping tests with model 2 were carried out in a similar way as those done with model 1 and the same set of test objects was used. The only differences were that new explorations were done with the model 2, and we were not as careful as before when localizing the test objects in different explorations. To simplify the test procedure the tactile information generated during the explorations of the test objects were written to and read from files. As with model 1 each test object was explored 5 times.

3.3 Results and Discussion

This model was able to discern all the objects and also to categorize them according to shape, i.e. the spheres were categorized in one area of the SOM, the blocks were categorized in another, and the cylinders were categorized in still another area of the SOM (Fig. 6). This might be compared with the discrimination ability of a child younger than 12 months. Such a child is able to discriminate course differences between objects placed in its hand, for example the child is able to discriminate between a cube and a cylinder but not between a cube and a cross-like stimuli (Millar, 2006).

4 Model 3

4.1 T-MPSOM

This model (Fig. 7) is similar to model 2 that uses the tensor product in several steps. The difference is that model 3 has substituted the tensor product operations with instances of a neural network. This neural network (T-MPSOM) is a variant of the SOM that allows multiple peak activations and receives two inputs. It is probably possible to generalize the idea to an arbitrary number of inputs.

There are several advantages when using the T-MPSOM instead of using the tensor product. One advantage is that the model becomes more biologically plausible. Another advantage is that the T-MPSOM is able to downscale the dimensions of the representation, which cannot be done with the tensor product. In model 3 the third instance of T-MPSOM consists of 1058 neurons, while the third instance of the tensor product module outputs a matrix with 270000 elements.

The T-MPSOM consists of a two-dimensional grid of neurons, where every neuron has two weight vectors corresponding to the dimensionality of the respective input vectors.

The sum of each input element multiplied with an arbor function (Dayan, 2000), which corresponds to the receptive field of the neuron, multiplied with the weight is calculated for both inputs. These sums are then multiplied with each other to yield the activity of the neuron.

All neurons in the neural network contribute to the updating of the weight vectors. The level of contribution from a certain neuron depends on the activity in the contributing neuron and a Gaussian function of the distance between the contributing neuron and the neuron with the weight considered for updating. This yields a multiple peak SOM. In every iteration, the input vectors and the weight vectors are normalized.

In mathematical terms, the T-MPSOM consists of a $i \times j$ matrix of neurons. In each iteration every neuron n_{ij} receives the two input vectors $a \in \Re^m$ and $b \in \Re^n$. n_{ij} has two weight vectors ${w_a}^{ij} \in \Re^m$ and ${w_b}^{ij} \in \Re^n$. The activity in the neuron n_{ij} is given by

$$x_{ij} = \sum_m A(i,m){w_a}^{ij}{}_m a_m \sum_n A(j,n){w_b}^{ij}{}_n b_n$$

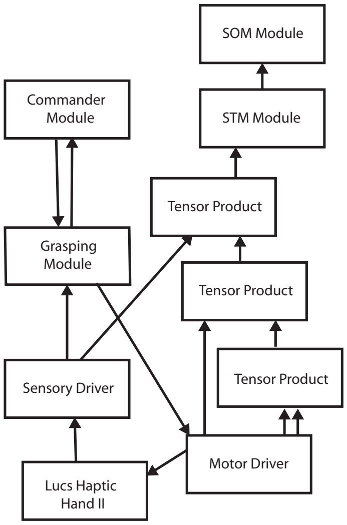


Figure 5: *Schematic depiction of model 2. The Lucs Haptic Hand II conveys the status of the sensors to the Sensory Driver, which in turn conveys the status of the sensors as an array to the Grasping Module and to the uppermost Tensor Product Module. The Motor Driver conveys, to the Lucs Haptic Hand II, the wanted positions of the servos in the robot hand and the wanted time to reach these positions. In addition, the Motor Driver conveys proprioceptive information, i.e. information about the configuration of the robot hand to the second Tensor Product Module. The Motor Driver also conveys to vectors to the lowermost Tensor Product Module. One of these vectors contains information about the wrist angle and the other contains information about the height of the robot hand. The lowermost Tensor Product Module conveys the resulting matrix of the tensor product between its to input vectors. The matrix is transformed into a vector by putting the rows after each other. The second Tensor Product Module conveys its output to the uppermost Tensor Product Module, again transformed to a vector with the matrix rows after each other. The Grasping Module conveys the wanted configuration of the robot hand and the wanted time to reach it to the Motor Driver, the status of individual grasps to the Commander Module. The uppermost Tensor Product Module conveys its output to the STM Module. The STM Module conveys a matrix of the tensor product superimposed from a whole exploration to the SOM Module. The Commander Module conveys orders to initiate grasping movements to the Grasping Module.*

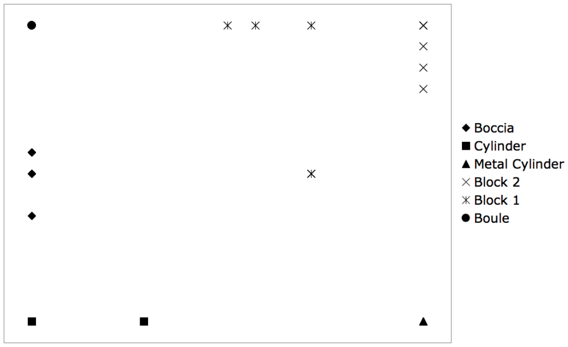


Figure 6: *Results of the grasping experiment with model 2. The depiction shows the centers of activity in the SOM during the testing with the training set patterns.*

in the variant with multiplied activations, and by

$$x_{ij} = \sum_m A(i,m){w_a}^{ij}{}_m a_m + \sum_n A(j,n){w_b}^{ij}{}_n b_n$$

in the variant with added activations, where

$$A(\alpha,\beta) \propto e^{-\left(\alpha - \frac{max\alpha}{max\beta}\beta\right)^2/2\sigma^2}.$$

The updating of the weight vectors are given by

$${w_a}^{ij}(t+1) = {w_a}^{ij}(t) - \alpha(t)\beta_{ij}(t)\left[a(t) - {w_a}^{ij}(t)\right]$$

and

$${w_b}^{ij}(t+1) = {w_b}^{ij}(t) - \alpha(t)\beta_{ij}(t)\left[b(t) - {w_b}^{ij}(t)\right],$$

where $0 \leq \alpha(t) \leq 1$, and $\alpha(t) \to 0$ when $t \to \infty$. The learning in each neuron is controlled by

$$\beta_{ij}(t) = \frac{\beta'_{ij}(t)}{max\beta'_{ij}(t)}$$

where

$$\beta'_{ij}(t) = \sum_k \sum_l x_{kl}(t) G(n_{kl}, n_{ij})$$

and $x_{kl}(t)$ is the activity in n_{kl} at time t and $G(n_{kl}, n_{ij})$ is a Gaussian function.

4.2 Grasping Tests

To simplify the test procedure with the model, the tactile information generated during the explorations of the objects in Table 1 were written to files. As before, 5 explorations were carried out with each object. The

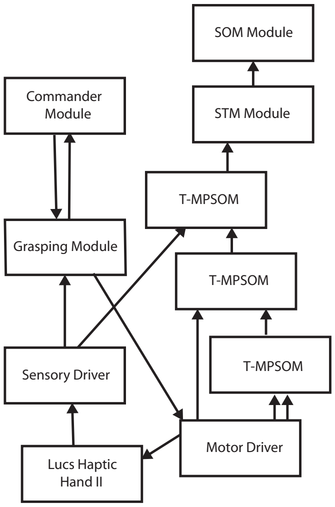


Figure 7: *Schematic depiction of model 3. The difference between this model and model 2 is that the Tensor Product Modules are replaced with T-MPSOM neural networks.*

model was first trained with randomly chosen explorations from the training set during, in total, 5000 iterations (one exploration consists of approximately 270 iterations). This was done to let the three instances of the T-MPSOM self-organize. When the learning had converged the model was exposed to each sample of the training set, i.e. in total 30 explorations and the output matrices from the STM Module were written to files. This set of 30 files was then used as a training and test set for the SOM Module. The model was trained with 1000 randomly chosen samples from its training set, after which the trained model was tested with all 30 samples in the training set.

4.3 Results and Discussion

The result from the grasping tests with this model (Fig. 8) is that the blocks, the cylinders and the spheres were categorized in different areas of the SOM. The model also separates the individual objects in the training/test set with only one exception when a boccia sphere was taken as a boule sphere. Therefore the capacity of this model is comparable to model 2 described above. We also experimented by varying the relationship between the input vectors and the size of the T-MPSOM in x and y directions. We found that the performance was superior when the relationship between the number of elements in input vector a and the number of neurons in the y direction

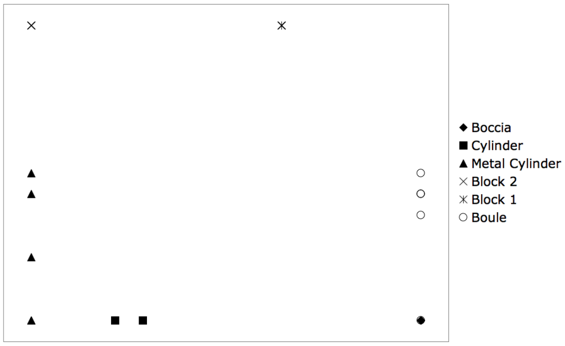


Figure 8: *Results of the grasping experiment with model 3. The figure shows the centers of activity in the SOM during the testing with the training set patterns.*

was similar to the relationship between the number of elements in input vector b and the number of neurons in x direction. This is probably due to that the activity is otherwise smeared out. Further we found that the sigma of the arbor function should be set rather small for the model to work well. The three instances of the T-MPSOM used 25, 90 and 1058 neurons. The SOM consisted of 225 neurons as in models 1 and 2.

5 Conclusions

We have designed and implemented three haptic models together with the Lucs Haptic Hand II. All three of them did, more or less, manage to categorize the test objects according to shape. Model 2, the one that used the tensor product in several steps, and model 3 with the novel T-MPSOM network worked best. These models are capable of both the categorization of the test objects according to shape, and to identify individual objects, with one exception for model 3, i.e. it once categorized a boccia sphere as a boule sphere. If model 3 had used a greater number of neurons it would probably not have been mistaking in any case. However, we avoided simulations with more neurons because we wanted a model that was not too computationally heavy, and the capacity of the current model appears to be comparable to that of a human. If the model were implemented in hardware a larger number of neurons would be acceptable. The worst performing model was model 1. That model did not perform very well in the categorization according to shape, and in addition it required a great accuracy in the location of a test object.

In the future we will further investigate the potential of T-MPSOM based models. In exchanging the STM and SOM modules to a SOM module with leaky integrator neurons we hope to obtain a haptic system consisting of artificial neural networks that self-organize in accordance with the input in all its parts. We will also experiment with a variation of the T-MPSOM that instead of multiplying add the activity contributions from each of the input vectors. Later we will study the interaction between haptics and vision.

Acknowledgements

We want to express our acknowledgements to Stiftelsen Elisabeth Rausings Minnesfond for finacial support for the Lucs Haptic Hand II, to Lars Kopp for the lifting mechanism of the robot hand and to Jacek Malec for helpful comments on the manuscript.

References

Arbib, M. A., Billard, A., Iacoboni, M., & Oztop, E. (2000). Synthetic brain imaging: grasping, mirror neurons and imitation. *Neural Networks*, 13, 975-999.

Balkenius, C., and Morén, J. (2003). From isolated components to cognitive systems. *ERCIM News, April 2003, 16*.

Coelho, J., Piater, J., & Grupen, R. (2001). Developing haptic and visual perceptual categories for reaching and grasping with a humanoid robot, *Robotics and Autonomous Systems*, 37, 2-3, 195218.

Dario, P., Laschi, C., Carrozza, M.C., Guglielmelli, E., Teti, G., Massa, B., Zecca, M., Taddeucci, D., & Leoni, F. (2000). An integrated approach for the design and development of a grasping and manipulation system in humanoid robotics, *Proceedings of the 2000 IEEE/RSJ international conference on intelligent robots and systems*, 1, 1-7.

Dario, P., Laschi, C., Menciassi, A., Guglielmelli, E., Carrozza, M.C., & Micera, S. (2003). Interfacing neural and artificial systems: from neuroengineering to neurorobotics, *Proceedings or the 1st international IEEE EMBS conference on neural engineering*, 418-421.

Dayan, P. (2000). Competition and Arbors in Ocular Dominance, *NIPS 2000*, 203-209.

DeLaurentis, K.J., & Mavroidis, C. (2000). Development of a shape memory alloy actuated robotic hand. (2004-10-28). http://citeseer.ist.psu.edu/383951.html

Fagg, A. H., & Arbib, M. A. (1998). Modeling parietal premotor interactions in primate control of grasping. *Neural Networks*, 11 (7 8), 1277-1303.

Heidemann, G., & Schöpfer, M. (2004). Dynamic tactile sensing for object identification, *Proceedings. ICRA '04. 2004 IEEE International Conference on Robotics and Automation*, 2004, 1, 813-818.

Hosoda, K., Tada, Y., & Asada, M. (2006). Anthropomorphic robotic soft fingertip with randomly distributed receptors, *Robotics and Autonomous Systems*, 54, 2, 104-109.

Jockusch, J., Walter, J., & Ritter, H. (1997). A tactile sensor system for a three-fingered robot manipulator, *Proceedings, 1997 IEEE International Conference on Robotics and Automation*, 1997, 4, 3080-3086.

Johnsson, M. (2004). Lucs Haptic Hand I - Technical Report, *Lucs Minor*, 8.

Johnsson, M. (2005). http://www.lucs.lu.se/People/Magnus.Johnsson/HapticPerception.html

Johnsson, M., Pallbo, R, & Balkenius, C. (2005a). Experiments with haptic perception in a robotic hand, *Advances in artificial intelligence in Sweden*, 81-86, Mälardalen University.

Johnsson, M., Pallbo, R., & Balkenius, C. (2005b). A haptic system for the Lucs Haptic Hand I, *Proceedings of IWINAC 2005*, 338-397, Springer Verlag.

Johnsson, M., & Balkenius, C. (2006a). Experiments with Artificial Haptic Perception in a Robotic Hand, *The Journal of Intelligent and Fuzzy Systems, in press*.

Johnsson, M., & Balkenius, C. (2006b). Lucs Haptic Hand II, *Lucs Minor*, 9.

Kohonen, T. (2001). *Self-organizing maps*, Berlin, Springer-verlag.

Millar, S. (2006). Network models for haptic perception, *Infant Behavior and Development*, 28, 3, 250-265.

Rhee, C., Chung, W., Kim, M., Shim, Y., & Lee, H. (2004). Door opening control using the multi-fingered robotic hand for the indoor service robot, *Proceedings of the 2004 IEEE international conference on robotics & automation, 4*, 4011-4016.

Sugiuchi, H., Hasegawa, Y., Watanabe, S., & Nomoto, M. (2000). A control system for multi-fingered robotic hand with distributed touch sensor, Industrial electronics society. *IECON 2000. 26th annual conference of the IEEE, 1*, 434-439.

Learning Anticipatory Behaviour Using a Simple Cerebellar Model

Harri Valpola[⋆]

[⋆]Laboratory of Computational Engineering
Helsinki University of Technology
P.O.Box 9203, FI-02015 TKK
Harri.Valpola@tkk.fi

Abstract

The cerebellar system is one of the best known components in the brain. Its computational principles are understood in sufficient detail to allow for using them for engineering applications. In this paper, some of the known facts about the cerebellar system are reviewed. The computational task of the cerebellar system appears to be prediction. On the other hand, the cerebellar system participates in motor control tasks. Feedback-error learning is one setting where a controller can be utilized in motor control. The limits of achievable temporal accuracy in feedback-error learning are investigated experimentally and it is shown that—somewhat surprisingly—motor output can have better temporal resolution than the error signal which is used for adapting the predictive controller. Finally, the algorithm is demonstrated in a simple but demanding simulated robot-control task.

1 Introduction

We still have plenty to learn from the brain. We would like to understand the underlying “algorithms” and computational principles and implement them in our artifacts. Not all details will be relevant but it can be expected that some general principles will turn out to be applicable, just as wings turned out to be a useful feature in airplanes but flapping was neither compatible with our technology nor necessary for flying.

Muscles are the major output of the brain (in addition to some hormonal regulation) and it is therefore fair to say that the brain has evolved to control movement. In order to understand the brain at a system-level, it is therefore useful to consider the problems encountered when controlling a body and interacting with the environment.

In this paper I will focus on anticipatory motor control implemented by the cerebellar system. It is one of the best understood parts of the brain both in terms of neurophysiology and computational principles. However, it seems that while this knowledge exists, it is not widespread because much of neuroscience research is focusing on neocortex in general and sensory processing in particular. Largely this is because it is easier to make well-controlled experimental setups but also because cortex is responsible for higher cognitive functions like planning, reasoning, attention and consciousness. Nevertheless, even in order to understand the neocortex, it is important to maintain a broader picture of the brain: there are many things that the neocortex does *not* need to do because other systems, like cerebellum, are taking care of them. The aim of this paper is to explain in simple terms what the cerebellar system does and demonstrate this in simulations.

2 Self-supervised learning

Thanks to extensive experimental, theoretical and engineering research, the cerebellar “algorithm” is nowadays well understood (see,e.g., Marr, 1969; Albus, 1971; Kawato and Gomi, 1992; Smith, 1998; Barto et al., 1999; Medina et al., 2000b,a; Medina and Mauk, 2000; Hansel et al., 2001; Hofstötter et al., 2002; Garwicz, 2002; Ito, 2002; Ohyama et al., 2003; Daniel and Crepel, 2003; Grethe and Thompson, 2003; Kawato, 2003; Apps and Garwicz, 2005; McKinstry et al., 2006). A brief summary of the research is that the cerebellar system turns out to be a predictor. In machine learning, predictors are usually adapted using supervised learning: there is a teacher who provides the correct answers and the task is to learn to mimic the answers provided by the teacher. Prediction tasks are about the only sensible supervised learning tasks in autonomous systems since there is no point of reinventing the wheel: if the teacher needs to be implemented anyway, there is no benefit in replicating the behaviour of the teacher. Supervised learning of prediction does make sense for an autonomous system because the system learns to provide the answers *before* the teacher gives the answers. Also, supervised learning is possible with a teacher who cannot give the correct answer but can give a correction. The student (or adaptive system) can therefore give faster and better answers than the teacher. This is pretty much the role of the cerebellar system: *the cerebellar system is specialized in making accurately-timed predictions in the timescale of about 100 ms – 5 s.*

Since the teachers are also located in the brain, one can talk about *self*-supervised learning. Cerebellar predictions are used for many purposes and correspondingly there are many types of teachers. In motor tasks, the teacher can be, for instance, a spinal or brain-stem reflex or it can be a focused, conscious decision of the neocortex. Cerebellum is implicated not only in motor tasks but in purely sensory and cognitive tasks, too (Allen et al., 1997). It is true not only for the cerebellar system but for many other adaptive modules in the brain that it is not the task but the algorithm that provides the most natural basis for understanding the modules. Nevertheless, motor tasks seem to be the most typical ones for the cerebellar system and they are also the focus in this paper.

3 Cerebellar system

Before going further into motor control, I will briefly introduce the anatomy of the cerebellar system and link it with the algorithm discussed above.

The cerebellar system consists of the inferior olive nucleus (IO), cerebellar cortex (CB) and deep nuclei (DN). Their computational roles are depicted in Figure 1. The teaching signals are relayed through IO (but are generated by diverse systems as noted earlier). The climbing fibres from IO deliver the teaching signal to CB and DN which implement the input-output mapping of the cerebellar system.

Roughly speaking, the cerebellar cortex is responsible for the accurate timing of cerebellar predictions but the deep nuclei implement the (linear) association between the inputs and outputs. Timing is implemented as disinhibition: normally CB inhibits DN but CB releases its inhibition at just the right moment to allow DN to activate its outputs (Perrett et al., 1993; Medina et al., 2000b). Note that when people talk about cerebellum, they usually refer to the cerebellar cortex, shown in Figure 2. This is because it is anatomically much larger and most of the computations seems to be going on there.

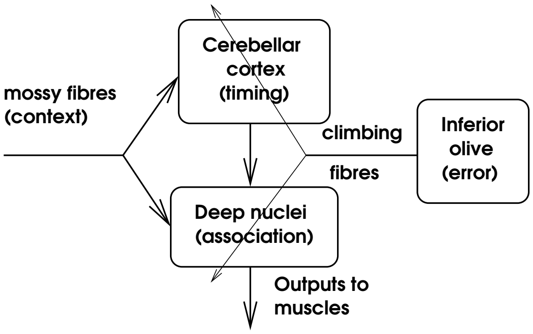


Figure 1: The "standard" architecture of the cerebellar system. In the part of cerebellum responsible for eye-movements, deep nuclei are missing from the loop and the cerebellar cortex is directly controlling brain stem.

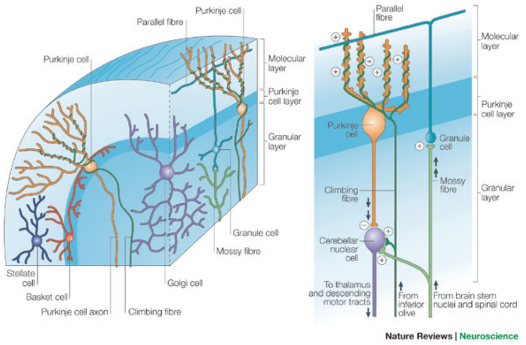


Figure 2: The anatomical structure and connections of the cerebellar cortex. The figure is reprocuded from (Apps and Garwicz, 2005).

4 Feedback-error learning and unknown delays

The "cerebellar algorithm" is supervised prediction which has no direct links with motor control and there are many ways in which a predictor could be used in motor control. Looking at the cerebellar system alone therefore does not tell much about the role cerebellar system plays in motor control. In fact, it is quite possible that there are multiple different uses. I will illustrate this with a gaze stabilization task.

When the visual images moves on the retina, it elicits an innate stabilizing reflex, the optokinetic reflex (OKR). There is a relatively simple wiring from retinal movement detectors to the eye muscles that compensate for the detected movement, making it smaller. In control engineering terms, this is a simple feedback regulator. The problem is that there is a significant delay between the compensatory control signals to eye muscles and sensory feedback from the signals. In control engineering this is called deadtime because from the viewpoint of the controller, the system appears to be unresponsive to any control signals.

Deadtime is considered to be one of the most difficult problems in control engineering. Consider how a feedback controller behaves in the face of deadtime: if the gain is high, the system starts oscillating wildly as a result of overcompensation that occurs during deadtime. If, on the other hand, the gain is low, the system is sluggish and doesn't properly compensate for errors—visual motion in this case.

After issuing a motor command, the controller should realize that a particular visual error has already been compensated for and that further compensatory movements are no longer necessary. This is clearly a task where a predictor might prove useful. There are at least two ways in which a predictor can be used:

Smith predictor: Predict the future state of the relevant sensory variable (such as visual motion in gaze stabilization) given a history of any relevant sensory and motor signals. Then use this predicted sensory variable as input to a feedback controller instead of the currently available (outdated) sensory input.

Feedback-error learning: Try to compensate for any errors before they are generated. Use the responses of a feedback controller as an error signal (not the target) for updating the motor predictions.

Roughly speaking, these two strategies are about sensory and motor prediction, respectively. The first strategy (alone) is called a Smith predictor after its inventor and the second one feedback-error learning because a feedback controller provides the error signal. These strategies can also be combined because sensory predictions can be useful inputs when making motor predictions.

The Smith predictor dates back to 1957 and is well-known and much studied in control theory. It can be quite sensitive to prediction errors which are bound to occur in real-world situations. In contrast, feedback-error learning seems to be more robust and a combination of the two where sensory prediction is used as an auxiliary variable for feedback-error learning should be able to combine the best of both worlds. In this paper I will concentrate on feedback-error learning.

In feedback-error learning, the predictor is supposed to predict and issue motor commands that silence the feedback controller. An obvious question that arises is how much in advance should the predictor try to issue the commands. Ideally this time would match the deadtime but it is usually difficult to know how long it is. If deadtime would be caused only by internal processing delays, it would at least be constant but unfortunately a significant portion of deadtime is often caused by inertia and other such properties of external world. These properties change from one task to another and correspondingly deadtime varies as well.

A simple solution is to spread the error signal temporally such that appropriate corrections are made at the optimal time in addition to corrections that are made at suboptimal times. On the one hand this solution could be considered as a feasible engineering solution: it will be more probable that the correct action will be taken at the correct time and in many cases it is more important to get the total amount of corrective movements right irrespective of their exact timing. On the other hand, it would be interesing to know how accurate timing the system can then learn? One might expect that the achievable accuracy depends on the amount of temporal spreading of the error signal. Rather surprisingly, this is not the case as will be demonstrated next.

Consider the following problem. There are 10 input channels $\mathbf{x}(t) = [x_1(t) \dots x_{10}(t)]^T$ which each carry information about the time. The n^{th} channel $x_n(t)$ in maximally active at time $t = n$. There is some overlap in the activations, in other words, the activations are spread temporally. In Figure 3, 10 input channels are active at times $1 \leq t \leq 10$ and are silent at times $11 \leq t \leq 20$. The task is now to find a linear mapping $y(t) = \mathbf{w}^T\mathbf{x}(t)$ of the inputs which outputs $y(t) = z(t)$ where the desired output $z(5) = 1$ and $z(t) = 0$ for $t \neq 5$.

If the error could be observed instantaneously $e(t) = z(t) - y(t)$, the problem could solved with standard supervised learning:

$$\Delta w \propto e(t)\mathbf{x}(t) \,. \tag{1}$$

In this example the deadtime is five time steps: $e(t) = z(t-5) - y(t-5)$. This is the signal that we receive from the teacher. If we knew the delay, we could simply match $e(t)$ with $\mathbf{x}(t-5)$ in (1), but now we assume that deadtime is unknown for the learning system and the error signal $e(t)$ will be matched with several $\mathbf{x}(t-\tau)$ with various different τ according to

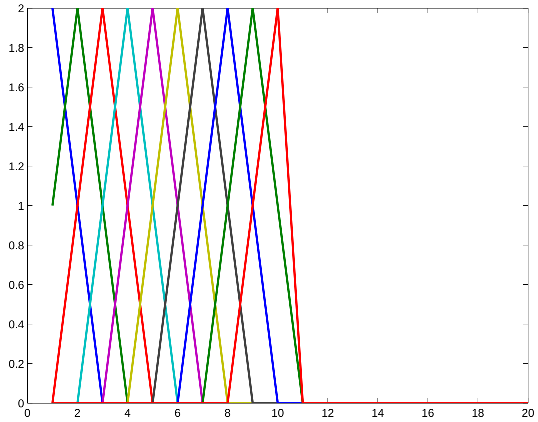


Figure 3: Ten input channels carry slightly overlapping information about the time.

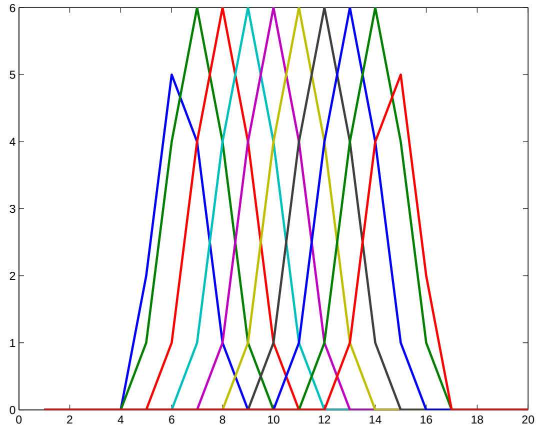


Figure 4: Eligibility traces which are achieved by delaying and temporally spreading the ten input channels in Figure 3.

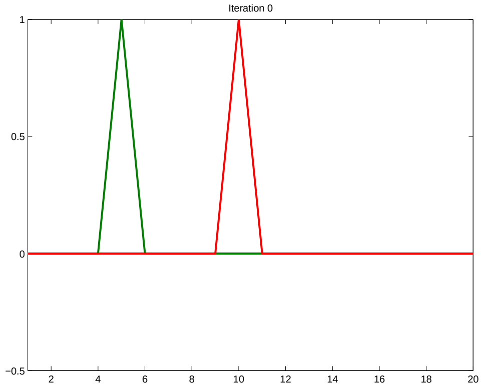


Figure 5: The weights are initialized to zeros and therefore the output (blue line behind other curves and doesn't show) is zero. The real error (green) occurs at time $t = 5$ but is observed (red) at time $t = 10$.

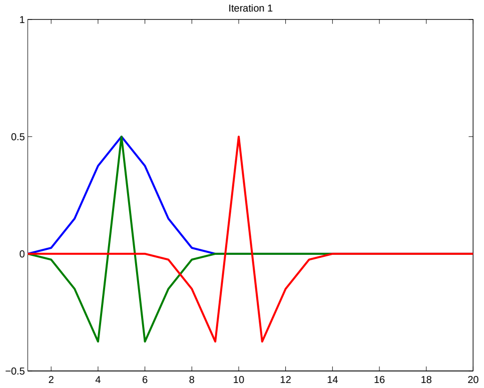


Figure 6: After the first weight update, the system responds (blue) around the correct time but temporal accuracy is low.

weighting kernel $h(\tau)$:

$$\Delta w \propto e(t) \int h(\tau)\mathbf{x}(t-\tau)d\tau\,. \qquad (2)$$

Note that in on-line learning system the matching needs to be implemented by delaying $\mathbf{x}(t)$ but conceptually this is equivalent to taking the error signal backward in time. The delayed inputs which are used in update rules are called eligibility traces. Figure 4 shows the eligibility traces of $\mathbf{x}(t)$ using delays of 4, 5 and 6 steps in proportions 1:2:1.

Figures 5–8 show how the response improves. The correct total amount of response is learned very quickly. Initially the response has poor temporal resolution which is expected because both the inputs and the error signal have poor temporal resolution. What is surprising is that after a large number of iterations the system converges to a response which is close to optimal.

In the above example the error signal was temporally centered at the correct delay and one might thus wonder whether convergence to optimal solution was simply due to the fortunate guess. This is not the case as shown by Figures 9–11. Although convergence is much slower, the iterations are clearly taking the system towards the correct solution. If 100,000 iterations sound like a lot, consider that people make about three saccades per second. Practice makes perfect.

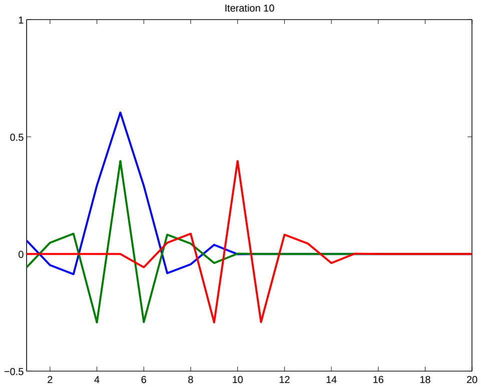


Figure 7: After 10 weight updates, the response has sharpened a bit.

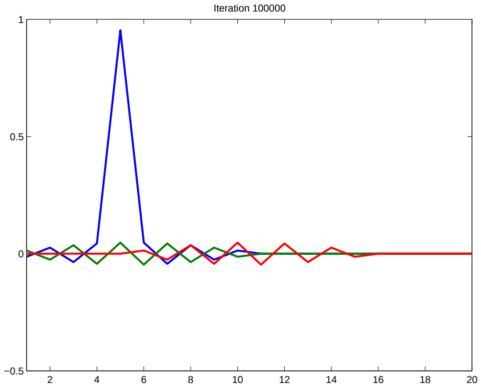


Figure 8: After 100,000 weight updates, the response is near optimal and certainly more accurate temporally than either the inputs or the error signal (after the spread).

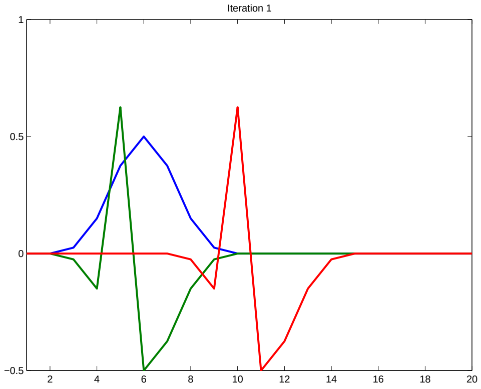


Figure 9: Delay is underestimated. After the first weight update, the system responds with low temporal accuracy around time $t = 6$.

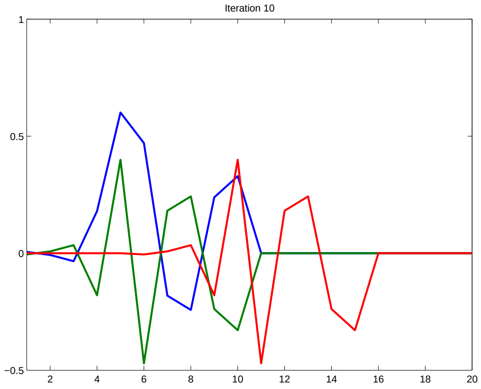


Figure 10: After 10 weight updates, the response has sharpened a bit and moved towards the correct delay.

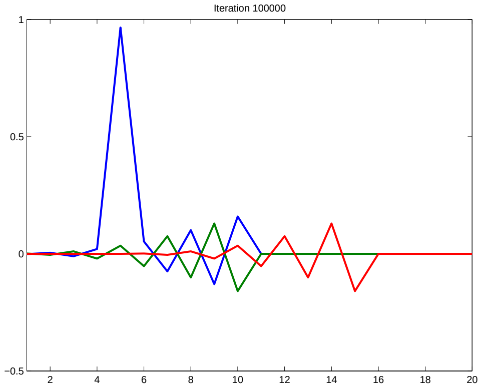


Figure 11: After 100,000 weight updates, the response is remarkably good although some amount of transient high-fequency ripple in the response is evident after $t > 5$. After around 300,000 iterations even that is almost gone.

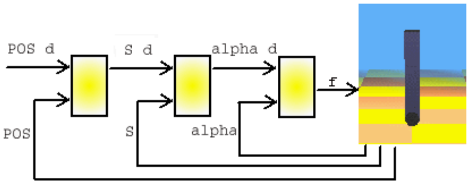


Figure 12: The “reflex” which provides the error signal for the predictive controller.

5 Robot simulations

I will next review experiments which demonstrate feedback-error learning in a simple but demanding robot control task. These simulations are taken from Heikki Joensuu’s master’s thesis (2006)[1] but similar work has been published for instance by Smith (1998) and McKinstry et al. (2006).

The robot is shaped like a pole and it moves in an arena by controlling its two wheels. In these simulations both wheels always had the same speed. Even though the robot cannot fall sideways, the body is quite unstable—quite like a unicycle. The robot was simulated using Webots[2] robot simulation environment.

5.1 Reflex

In feedback-error learning, the task of the controller is given in terms of a “reflex” or a feedback controller which gives a corrective motor signal, the error $e(t)$ in (2), if some sensory input level is non-optimal. In this simulation, the task of the robot is to stay in the middle of the arena in upright position. This is expressed in terms of a simple hierarchical P-controller (Figure 12) where each P-controller compares its desired value of a variable (reference value in control engineering terms) with the observed value and issues a control signal which is directly proportional to the difference (P stands for proportional). At the highest level is a position controller which outputs a desired speed in proportion to displacement from the center of the arena. The speed controller outputs a desired tilt angle for the robot: if the speed is optimal, stay upright; if the speed is too low, lean forward; and so on. The last controller tries to achieve the desired tilt angle by controlling the acceleration of the wheels.

Note that in this simulation, there were practically no internal processing delays. Due to inertia of the body, the controller will nevertheless need to anticipate. Although there is strictly speaking no deadtime, inertia works in much the same way and internal processing delays can therefore be handled by the same system. Also note that if the body changes—grows, for example—inertia changes and hence the optimal target advance in prediction changes, too.

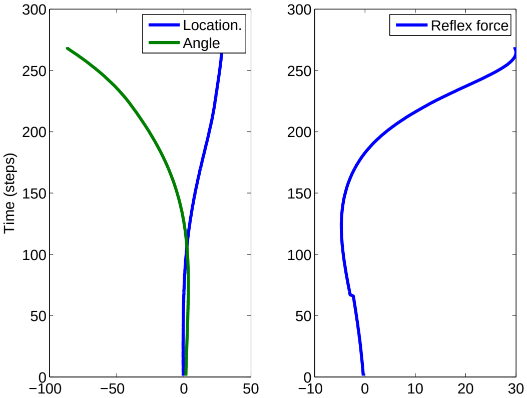


Figure 13: Behaviour with the reflex alone.

5.2 Results

As expected, the reflex alone cannot keep the robot up (Figure 13) but it can give useful corrections which can be used by a predictive controller to update its predictions. The target anticipation was on average 40 time steps and it was temporally spread. After 160 trials (Figure 14), the robot is able balance itself but does not succeed in centering itself (this is very typical behaviour during learning). After 300 trials (Figure 15), the robot can both balance and center itself rapidly and accurately.

In the above simulations, the input for the cerebellar model (the linear mapping described in Section 4) was “proprioceptive”, concerning only its own body. A convenient property of predictors is that they can take almost any type of inputs; any information that will turn out useful will be used for making the predictions. Figure 16 depicts the behaviour of the robot after it has learned to anticipate a ball hitting it. The only change that needed to be made to the system was to include two input channels, one for balls approaching from the left and another for balls from the right. With more sensory inputs the controller could even learn to deal with balls of different properties.

[1] Videos and other material related to the thesis are available at `http://www.lce.hut.fi/research/eas/compneuro/projects/cerebellum/`.

[2] `http://www.cyberbotics.com/`

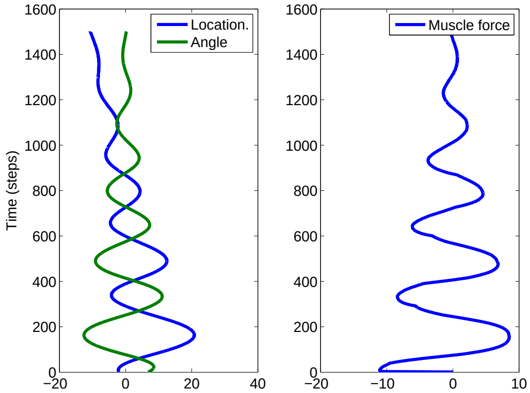


Figure 14: Behaviour after 160 trials.

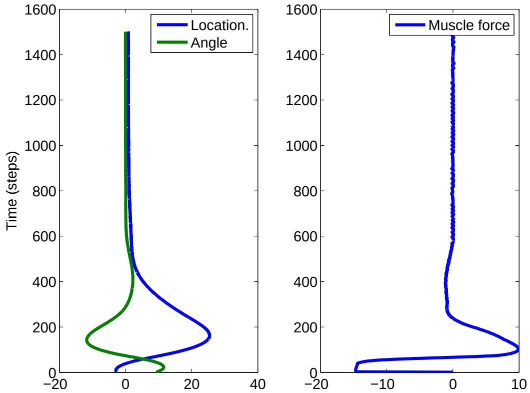


Figure 15: Behaviour after 300 trials.

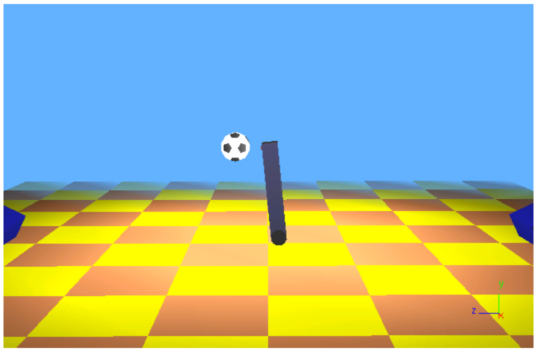


Figure 16: The robot has learned to make anticipatory movements before the ball hits it and is therefore able to cope with the impacts.

6 Discussion

The simulations discussed in this paper had just one degree of freedom while our own cerebellum controls hundreds of muscles. An interesting question is how this control generalizes to multiple degrees of freedom. I think that a feasible strategy for learning coordinated movements is to have independent reflexes for each degree of freedom (or reflexes that each control at most a few degrees of freedom) and to let the cerebellar model learn from experience how the movements in one part of the body need to be compensated by another part. This prediction task will be easier if the predictor for each degree of freedom gets information about the ongoing control signals sent to all other degrees of freedom.

Since the role of cerebellum in motor control is well understood, we can ask what should other parts of the brain do to help cerebellum achieve its job. One thing that cerebellum clearly needs is information which is useful as input for the predictor. For instance, if the mass distribution of the pole changes, the optimal control changes. It would therefore be useful to have system that monitors the dynamic properties of the body. In mammals, neocortex provides information about the state of the world, including the state of ones own body and its changing dynamic properties, intentions and goals, all of which are useful inputs for cerebellum to accomplish its control tasks.

One thing that the cerebellar system clearly cannot do is acquire new goals and reflexes. Not all goals can be predefined in terms of hard-wired reflexes and it is therefore necessary to have systems that learn new reflexes or otherwise generate corrective movements that lead to rewarding consequences. In mammals, basal ganglia and prefrontal cortex are known to collaborate in the generation of such goal-directed behaviour.

In conclusion, the “cerebellar algorithm”—prediction—is well understood on a general level. There are many (mutually compatible) ways in which a predictor can be used in control. In this paper, I focused on feedback-error learning. I showed that the temporal accuracy that can be achieved in such a strategy is surprisingly good, better than the temporal accuracy of the error signal. A used a simulated robot balancing task to demonstrate that feedback-error learning is, indeed, able to learn to control the unstable robot far better than the reflex that generates the error signals. The algorithm is computationally efficient and it is easy to use it as a module in a larger system.

Acknowledgements

I would like to thank M.Sc. Heikki Joensuu, who implemented self-supervised learning and the controllers for the robot in Matlab, and Mr. Harm Aarts, who developed the interface between Webots robot simulator and Matlab.

References

J. S. Albus. A theory of cerebellar function. *Mathematical Biosciences*, 10:26–61, 1971.

G. Allen, R. B. Buxton, E. C. Wong, and E. Courchesne. Attentional activation of the cerebellum independent of motor involvement. *Science*, 275:1940–1943, 1997.

R. Apps and M. Garwicz. Anatomical and physiological foundations of cerebellar information processing. *Nature Reviews Neuroscience*, 6:297–311, 2005.

A. G. Barto, A. H. Fagg, N. Sitkoff, and J. C. Houk. A cerebellar model of timing and prediction in the control of reaching. *Neural Computation*, 11:565–594, 1999.

H. Daniel and F. Crepel. Cerebellum: Neural plasticity. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 196–200. The MIT Press, 2nd edition, 2003.

M. Garwicz. Spinal reflexes provide motor error signals to cerebellar modules — relevance for motor coordination. *Brain Research Reviews*, 40(1):152–165, 2002.

J. S. Grethe and R. F. Thompson. Cerebellum and conditioning. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 187–190. The MIT Press, 2nd edition, 2003.

C. Hansel, D. J. Linden, and E. D'Angelo. Beyond parallel fiber ltd: the diversity of synaptic and nonsynaptic plasticity in the cerebellum. *Nature Neuroscience*, 4:467–475, 2001.

C. Hofstötter, M. Mintz, and P. F. M. J. Verschure. The cerebellum in action: a simulation and robotics study. *European Journal of Neuroscience*, 16(7):1361–1376, 2002.

M. Ito. Historical review of the significance of the cerebellum and the role of purkinje cells in motor learning. *Annals of the New York Academy of Sciences*, 978:273–288, 2002.

H. Joensuu. Adaptive control inspired by the cerebellar system. Master's thesis, Helsinki University of Technology, Finland, 2006.

M. Kawato. Cerebellum and motor control. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 190–195. The MIT Press, 2nd edition, 2003.

M. Kawato and H. Gomi. The cerebellum and VOR/OKR learning models. *Trends in Neuroscience*, 15(11):445–453, 1992.

D. Marr. A theory of cerebellar cortex. *Journal of Physiology (London)*, 202:437–470, 1969.

J. L. McKinstry, G. M. Edelman, and J. L. Krichmar. A cerebellar model for predictive motor control tested in a brain-based device. *Proc Natl Acad Sci USA*, 103:3387–3392, 2006.

J. F. Medina, K. S. Garcia, W. L. Nores, N. M. Taylor, and M. D. Mauk. Timing mechanisms in the cerebellum: Testing predictions of a large-scale computer simulation. *The Journal of Neuroscience*, 20 (14):5516–5525, 2000a.

J. F. Medina and M. D. Mauk. Computer simulation of cerebellar information processing. *Nature Neuroscience*, 3:1205–1211, 2000.

J. F. Medina, W. L. Noresa, T. Ohyama, and M. D. Mauk. Mechanisms of cerebellar learning suggested by eyelid conditioning. *Current Opinion in Neurobiology*, 10(6):717–724, 2000b.

T. Ohyama, W. L. Nores, M. Murphy, and M. D. Mauk. What the cerebellum computes. *Trends in Neurosciences*, 26(4):222–227, 2003.

S. P. Perrett, B. P. Ruiz, and M. D. Mauk. Cerebellar cortex lesions disrupt learning-dependent timing of conditioned eyelid responses. *Journal of Neuroscience*, 13:1708–1718, 1993.

R. L. Smith. *Intelligent Motion Control with an Artificial Cerebellum*. PhD thesis, University of Auckland, New Zealand, 1998. URL `http://www.q12.org/phd.html`.

Heuristics for Co-opetition in Agent Coalition Formation

Kevin Westwood
Utah State University
kevwestwood@cc.usu.edu

Vicki H. Allan
Utah State University
vicki.Allan@usu.edu

ABSTRACT

Coalitions are often required for multi-agent collaboration. In this paper, we consider tasks that can only be completed with the combined efforts of multiple agents using approaches which are both cooperative and competitive. Often agents forming coalitions determine optimal coalitions by looking at all possibilities. This requires an exponential algorithm and is not feasible when the number of agents and tasks is large. We propose agents use a two step process of first determining the task and then the agents that will help complete the task. We describe polynomial time heuristics for each decision. We measure four different agent types using the described heuristics. We conclude that if agents only consider their own individual gain, the potential profit of the agents and the potential throughput of the system will not be realized.

1 Introduction

Co-opetition is a phrase coined by business professors Brandenburger and Nalebuff (1996), to emphasize the need to consider both competitive and cooperative strategies. Recently there has been considerable research in the area of multi-agent coalitions (Lau and Lim, 2003) (Lau and Zhang, 2003)(Belmonte, et. al, 2004) (Li and Soh, 2004) (Lau and Zhang, 2003) (Blankenburg, Klusch, and Shehory, 2003) (Kraus, Shehory, and Taase, 2004). Multi-agent coalitions have been applied to many areas including manufacturing for optimizing production (Dang, et.al., 2003), information searching in heterogeneous databases (Klusch and Shehory, 1996), and, most recently, the area of e-commerce (Li, et.al., 2003).

A *coalition* is a set of agents that work together to achieve a mutually beneficial goal (Klusch and Shehory, 1996). The formation of a collation is often required because an agent is unable to complete a task alone. However, there may be other motivating forces, such as increasing its utility or completing the task by a deadline.

In the request for proposal (RFP) domain (Kraus, Shehory, and Taase, 2003), a collection of agents is challenged to complete specific tasks, each of which can be divided into subtasks. In this format, the task requirements and utility are specified and requests for proposals from agents are sought.

In the scenario considered here, no single agent possesses all the skills necessary to complete a task, and so coalitions must be formed. The agents attempt to form coalitions that will be able to complete tasks in a way which maximizes the payoff received. The agents are *individually rational* and will not join a coalition unless participating in the coalition is better than not joining a coalition at all. However, agents use heuristics to facilitate accepting less than optimal personal benefit if the global utility can be increased – hence the term, co-opetition.

2 Skilled Request for Proposal Domain

We use the term *Skilled Request For Proposal* (SRFP) to denote a domain in which levels of skill play an important role. In our domain, a requester provides a set of tasks $\mathcal{T}$ = $\{\mathcal{T}_1 \ldots \mathcal{T}_n\}$ to be completed. Task i is divided into a list of subtasks $\mathcal{T}_i = (t_{i1} \ldots t_{im})$. Each subtask t_{ij} is associated with a skill-level pair $(\mathcal{ts}_{ij}, \mathcal{tl}_{ij})$ where a task skill, $\mathcal{ts}$, represents a skill taken from a skill set $\mathcal{S}$, i.e., $\mathcal{ts} \in \mathcal{S} = \{s_1 \ldots s_k\}$ and $\mathcal{tl}$ represents a specific level of skill, $\mathcal{tl} \in \{1 \ldots 10\}$. A set of service agents $\mathcal{A} = \{A_1 \ldots A_p\}$ exists such that each agent k has an associated fee, f_k, and agent skill-level pair $(\mathcal{as}_k, \mathcal{al}_k)$. The payoff value associated with a task is $V(T_i)$.

A coalition $\mathcal{C}_i = <\mathcal{T}_i, C_i, \mathcal{P}_i>$ consists a task vector $\mathcal{T}_i = (t_{i1} \ldots t_{ij} \ldots t_{im})$ for task i, consisting of each of the subtasks (indicated by *j*) required to perform the task, a vector of agents comprising the coalition for task i $C_i=(C_{i1}\ldots C_{ij}\ldots C_{im})$ (where $C_{ij}\in \mathcal{A}$, and the elements of the tuple are ordered to indicate C_{ij} performs subtask t_{ij}), and a payment vector $\mathcal{P}_i = (p_{i1}\ldots p_{ij}\ldots p_{im})$ in which $p_{ij} \geq f_{ij}$ for all agents C_{ij} with skill-level $(\mathit{as}_{ij},\mathit{al}_{ij})$ and fee f_{ij} . Since the system is pareto optimal, all utility is distributed, i.e., $\sum_j p_{ij}=V(T_i)$. Note that, f_{ij} represents the actual cost associated with agent i performing the jth subtask. A rational agent will not consent to performing a subtask for less than it's actual cost.

To facilitate the discussion, we introduce some terminology. The task $T_i = (t_{i1}\ldots t_{im})$ is *satisfied* by the coalition consisting of agents $C_i=(C_{i1} \ldots C_{id} \ldots C_{im})$ $(C_{id}\in \mathcal{A})$ if each agent d of the coalition (C_{id}) has associated skill-level pair $(\mathit{as}_{id},\mathit{al}_{id})$ such that $\mathit{as}_{id} = \mathit{ts}_{id}$ and $\mathit{al}_{id} \geq \mathit{tl}_{id}$. In other words, a coalition satisfies the needs of a task if for each subtask, there is an agent in the coalition with the required skill at a level equal to or greater than the level specified for the subtask. The difference $\mathit{al}_{id} - \mathit{tl}_{id}$ is termed the *underutilization* of the dth agent in performing task i.

We term a task *viable* if there are available agents to satisfy the requirements of the task. We say an agent *supports* a subtask if the agent can perform an (as of yet) unassigned subtask. A level equal to or greater than the required level is termed an *acceptable level.* The level of skill which is actually required (due to possible missing skill-level pairs in the available task set) is called the *effective level.* Note, the effective level is the minimum of all acceptable levels (for a skill) in the current agent population.

For a particular skill, the base fees are a non-decreasing function of level and are publicly known. In our domain, while the specific fee associated with an agent is peculiar to the agent, the set of fees for a given skill-level pair is taken from a normal distribution with known mean and standard deviation.

3 Previous Work

Lau and Zhang (2003) explore the complexity of exhaustive methods for coalition formation.

The auction protocol proposed by Kraus, Shehory, and Taase (2003) is conducted by a central manager. Agents are randomly ordered at the beginning of each round r = {0,1,2…}. In each round, the agents are asked (in turn) whether they want to respond to previous proposals or initiate a proposal of their own. Each agent gets exactly one turn per round. After every agent has completed its turn in a round, for each satisfied proposal, the associated task and the agents required to complete the task are removed from the auction. At the end of a round, any unsatisfied proposals are discarded. The payoff associated with a task is decreased by a fraction of its original value with each round. Thus, for a task completing in round r with *discount factor* δ, the payment value is $V(T_i)^* \delta^r$. An initiating agent may only propose to agents which follow it in the non-cyclic agent ordering. This limit placed on who can be involved in a proposal makes the protocol unrealistic in modeling real world coalition formation. This limit is not present in our system. Our work also differs from their model in that there is no discount factor. The only motivation for forming coalitions early in our domain is that there are a greater number of tasks and agents from which to select.

In the domain proposed by Kraus, Shehory, and Taase (2003), each subtask is completely unique from all other subtasks encountered. A boolean function $\Phi(A_k, t_{ij})$ determines whether agent A_k can perform subtask t_{ij}. An agent has a 40% chance of having the skills required to complete a "normal task" and a 15% chance of having the skills to complete a "specialized task". This skill structure is limited in its ability to model realistic skill structures as there is only a rough concept of supply and demand on an individual skill basis. There is no concept of using a more highly skilled agent to do a less demanding task. There is also no concept of paying more for a higher level of skill. Kraus, et.al., consider incomplete information, but because averages are known, incomplete information provides merely random changes in the overall behavior, increasing the variance but not changing the cumulative behavior. Compromise is introduced in (Kraus, Shehory, and Taase, 2004) to increase the speed at which coalitions can be formed.

Gerber and Klusch (2003) worry about the ability to create stable coalitions in a dynamic environment where it is not reasonable to compute kernel-stable coalitions each time the state changes due to the high complexity. In addition, an agent's reputation for completing a task (reliability) enters into coalition formation decisions. Interestingly, an agent who attempts to form a coalition (which is later rejected) is penalized in terms of its reliability rating if a substitute coalition can not be formed. Thus, to protect its reliability rating, an agent who has received commitments from some agents must attempt to utilize the partial coalition in proposing a new task. Potentially overlapping coalitions are allowed and must be evaluated, increasing the complexity of heuristics that require coalitions to be disjoint. Similarly, Caillou, Aknine, and Pinson (2002) are concerned with dynamic restructuring of coalitions.

Scully, Madden, and Lyons (2004) use a skill vector that describes speed, and reliability. We propose a combined skill-level pair. Blankenburg, Klusch, and Shehory (2003) introduce the idea forming coalitions when there is a fuzzy knowledge of various information. An agent's evaluation of the coalition is uncertain, and agents can be (fuzzy) members of multiple coalitions to various degrees.

Trust based coalition formation as discussed in (Blankenburg, et.al., 2005) to deal with agents who deliberately misrepresent their costs and coalition valuation to increase their own profits. Belmonte et.al. (2004) allow agents to perform more than one task at a time, and an agent can break an initial agreement by paying a transfer cost. To simplify the tests and comparisons in the model, in the SRFP each agent has a single skill $\mathcal{as}$ in the skill set $\mathcal{S}$ = {s_1 … s_k}, but multiple skills could be modeled by storing a set of skills for each agent.

4 Auctioning Protocol

The coalition protocol described in Kraus, Shehory, and Taase (2003) is used as a simplification of a parallel model. We employ a similar protocol, consisting of service agents and a central manager that acts as an auctioneer and coalition manager. Service agents and tasks can join the auction at any time. For simplicity, tasks and agents are not allowed to leave the auction.

The auction is a variation of an English auction. The manager has knowledge of a set of tasks T = {T_1,…,T_n}. Each task has a value V(T_i) that will be paid when the task is complete. The auctioneer posts all tasks and their associated value to be paid. The auction consists of rounds {0,…,r_z} that will continue until all tasks are auctioned off, no remaining task is viable, or the time allotted expires. In order to focus on the ability of the heuristics to aid coalition formation, no discount factor is applied.

All negotiation is done through the manager. At the beginning of the auction, the manager randomly orders all agents so as not to give any preference to particular agents. One by one, each agent can either propose a coalition to perform a task or accept a proposal another agent has already made. If agent k chooses to make an offer, the agent is termed the *initiating agent* for the coalition. An agent who chooses to accept a previous offer is termed a *participating agent*. A proposal specifies the task to be done and the set of agents to complete the task. Profits are divided proportionally. After agent k has decided whether to propose a task or accept a previous proposal, the turn passes to agent k+1 (mod p) (who decides to propose or accept a previous offer). All agents accept or reject the proposal only when it is their turn in the round.

Consider the life a proposal initiated by agent k. All agents of the proposed coalition are informed of the proposal at their turn. They can accept or reject. If any agent rejects, when the turn passes to agent k in the next round (after all other agents have had a turn), the agents who agreed to the proposal are informed that the proposal failed. Agent k is allowed to make another proposal, re-propose the same proposal, or accept a previous proposal. If all the agents accept the proposal, when the turn passes to agent k, the task is awarded to the participating agents, and the task and participating agents are removed from the auction.

When an agent *i* is deciding between accepting one of the current offers or making an offer of its own, it has no knowledge of proposed coalitions which do not involve *i* nor does it know how many agents have accepted or rejected the current proposals. This lack of current information reflects the state of information in a parallel proposal environment in which acceptances happen simultaneously. Since agents and tasks are removed from the auction when the initiating agent has a turn in the following round, the information is updated with predictable delay. For the current offers, the task and the net gain (given the proposed coalition) is known.

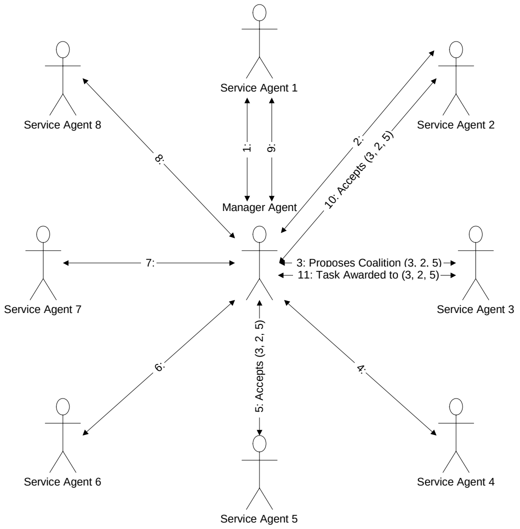


Figure 1: Auction Process

When a task is completed, each agent k is paid their fee f_k plus a portion of the *net gain* of the coalition.

The net gain is calculated as $\mathcal{N}_i = V(T_i) - actualCost_i$, where $actualCost_i = \sum_{j=1}^{m} f_{ij}$. In other words, the net gain is the amount that is left over from the value of the task after the agents are paid their advertised fees. Net gain is divided among the agents. Historically, the portion paid to each agent is divided equally, proportionally according to the agent's individual cost, or differentially, based on demand for a particular skill. In our model, we divide the net gain proportionally. Thus, agent i receives $f_{ij} + (f_{ij}/actualCost_i)^* \mathcal{N}_i$.

5 Agent Algorithms

In many coalition formation algorithms, proposing agents calculate all possible coalitions to complete all tasks, an exponential algorithm. Because this quickly becomes unreasonable for even moderately sized problems, we divide the problem into two components and propose the following polynomial time heuristics for each component. In the first component, the agent selects a task to complete. In the second component, the agent selects the set of agents for the coalition to complete the task.

In this paper, we explore four task selection heuristics and two coalition selection heuristics for the proposing agents.

5.1 Task Selection Heuristic

The task selection heuristics are as follows:

1) **Individual Profit Task Selection:** An agent selects a task that will maximize its own profit. For each task, the agent first computes the expected net gain by subtracting the average fee of the effective level of each required skill. Thus, knowledge of the available agents is used to more accurately predict the agents' fees for completing a task. Tasks are ranked by comparing the agent's individual utility in each possible task. The task which yields the maximum individual utility is selected.
2) **Global Profit Task Selection:** An agent selects a task that will maximize global profit. For each task, the agent computes the expected net gain by subtracting the average fee of the effective level of required skill. The task with the maximum net gain is selected.
3) **Best Fit Task Selection:** An agent picks the task for which the sum of the agent underutilization is minimized. The motivation is that when the resources are used wisely future coalitions will be less likely to face resource constraints. Ties are broken using maximum individual profit.
4) **Co-opetitive Selection:** Select the best fit task as long as it is within P% of the maximum individual profit.

5.2 Coalition Selection Heuristic

The coalition selection heuristics are as follows:

1) **Best Profit Coalition Selection:** For each subtask, select the supporting agent with the smallest fee. Since actual costs differ from the expected costs, this is different than merely finding the best fit.
2) **Best Fit Coalition Selection:** For each subtask, select the supporting agent with the smallest underutilization. In general, a closer match of level will conserve resources while getting the best expected cost.

5.3 Acceptance Policy

Agents must have a policy to decide between (a) accepting one of the various proposals or (b) being an initiating agent. Our agents compare existing proposals with the benefits of the proposal they would make (using their task selection heuristic) and select the proposal most closely fitting their criteria. The *contending proposal* for agent i is the "best" proposal (of those proposed to agent i)

Accepting the contending proposal may give the agent a better chance of joining a coalition, particularly when the number of subtasks is small, as an agent knows that at least two agents (itself and the initiator) will accept the proposal. Since an agent i has no information about other proposals (not involving i) and no knowledge of how many agents have accepted the current proposal, the advantage of accepting a current proposal over initiating a proposal is not as great as one might think.

If the contending proposal is within C% of the proposal the agent would initiate, the contending proposal is accepted. Otherwise the proposal is rejected. Several different values for the compromising ratio, C, are evaluated in our experiment. The idea is that with compromising (not holding out for the best possible coalition) coalitions can be formed quicker, resulting in higher utility (attributed to more opportunity due to a richer pool of tasks and agents) (Kraus, Shehory, and Taase, 2004).

6 Results

Results represent the average of tests run 5000 times. Each trial involves 60 agents and 20 tasks (3 subtasks each). Each subtask has a randomly generated agent skill-level pair. Skills vary from 1 to 10 and skill levels vary from 1 to 10. Agent skills and skill levels are matched to fit the tasks. For example, if subtask 1 from Task 1 has an agent skill-level pair of (3,5), an agent is generated with skill 3 level 5. This setup gives an environment where most tasks can be assigned and patterns among the agents can easily be discovered.

For each skill, skill level 1 has a base fee of 5 and skill level 10 has a base fee of 50. Actual fees are assigned using a normal distribution with a median value of the base fee and a standard deviation of 2.5 (half the difference from one skill level to the next). In our tests, payoffs of the generated tasks are assigned as a random percentage (100-200%) of the expected cost of completing the task (based on expected costs of the skill-level required by the subtasks).

Four different agent types are considered, corresponding to the task selection algorithms: Local Profit, Global Profit, Best Fit and Co-opetitive. Local Profit and Global Profit agents use the best profit coalition selection algorithm. Best Fit agents use the best fit coalition selection algorithm. The Co-opetitive agents use the best fit coalition selection algorithm if the best fit task's potential individual profit is greater than 90% of the best profit task's potential individual profit. Otherwise, the Co-opetitive agent uses the best profit coalition selection algorithm. The Co-opetitive agent is interested in helping the throughput of the system, if individual benefit is considered. Thus, a high P% was chosen. For our tests, we use a P value of 90.

All profits are shown as Profit/Optimal profit or the *Profit Ratio*. This representation is important as it normalizes profits from one agent to another. Consider agent A that has an optimal profit of 100 and agent B that has an optimal profit of 10. If agent A receives 50 and agent B receives 5, each agent's performance is equal. Raw results tend to exhibit no clear pattern of behavior as the achievable results are so variable.

Individual optimal profits are determined by exploring all possibilities for the specific agent. Global optimal profits are determined by using a hill-climbing algorithm.

6.1 Compromising Ratio

We need to know what degree of compromise is affective in determining an acceptance policy. We hypothesized that the more competition an agent has or the fewer the number of tasks that are available to complete, the more the agent would want to compromise in accepting other agents' proposals. If the agent does not compromise, it risks the chance of not joining a coalition.

To determine the compromising ratio, we conduct the following experiment. The number of tasks is fixed at 20 with 3 subtasks each. The number of agents fluctuates from 20 to 180. Using an equal distribution of each agent type, we track the individual profit and the number of tasks completed by each agent type.

Figure 2 shows the profit ratio for the Individual Profit agent. On average, having a high compromising ratio maximizes the Individual Profit agent's profit. This same pattern is seen in other agent types as well. The number of tasks that are completed is maximized by Best Fit agents having a high compromising ratio. Because of these results, we fix the compromising ratio at 90% for all agent types in the following tests.

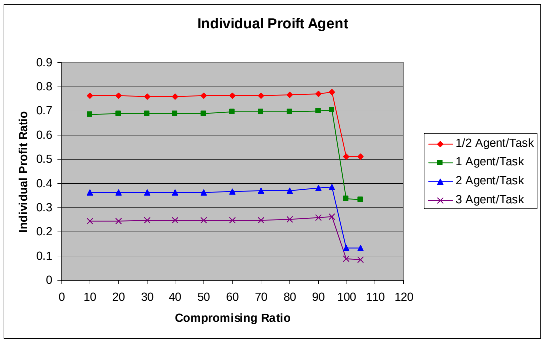


Figure 2: Individual Profit Agent and Compromising Ratio

6.2 Individual Agents

Each agent type has a different objective. The Individual Profit agent desires to increase it's own individual profit. The Global Profit agent desires to maximize the profit of all agents in the system. The Best Fit agent desires to maximize the throughput of the system. The Co-opetitive agent desires to increase the throughput of the system as long as it is still benefiting individually.

We explore the impact each agent type has on the system. We setup tests that gradually increase the percent of a single agent type until it is the only agent type left in the system. Note that the other agents' types in the system exist in equal proportions.

Figure 3 shows the individual profit ratio for the specific agent type as its percent in the system increases. It can be seen that the Individual Profit agent does very well when less than 70% of the agents are Individual Profit Agents. As its percentage in the system increases, their profit decreases. The primary cause of this decrease is that there are fewer tasks completed. See Figure 4.

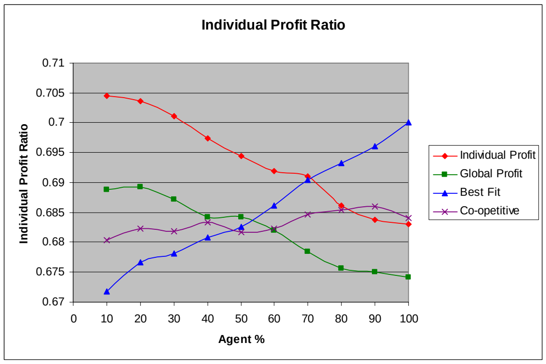


Figure 3: Individual Profit Ratio

The Best Fit agents have the opposite effect. As the percentage of Best Fit agents increase, the number of tasks completed and the individual profit for the Best Fit agents increase as well.

The Co-opetitive agents' profit increases slightly, but is fairly steady. With less than 50% of the same agent type, the Co-opetitive agent has a higher individual profit than the Best Fit agent. The throughput of the Co-opetitive agent is fairly steady as well. As the percentage of Co-opetitive agents increases, the average number of tasks completed stays about the same.

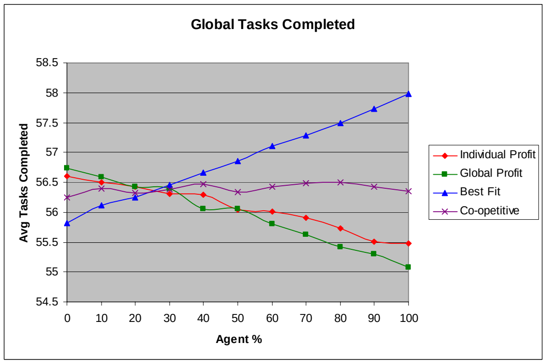


Figure 4: Global Tasks Completed

Figure 5 shows the global profit ratio of the system. This is calculated as the total profit of all agents divided by maximum possible profit of all agents. The Global Profit agents' objective was to maximize the global profit and in doing so increase the throughput of the system. It did not do as well as hypothesized.

The Global Profit is dependent on the number of tasks that are completed. The Best Fit agent maximizes the global profit as its percentage increases past 25%. Both profit agents do not do very well. As their percentage increases, the global profit decreases. The Co-opetitive agent does a little better than the profit agents, but not nearly as well as the Best Fit agent.

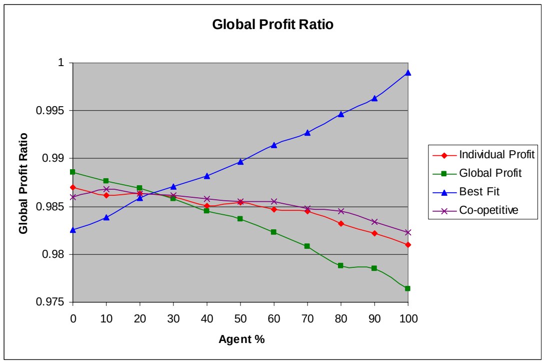


Figure 5: Global Profit Ratio

6.3 Standard Deviation

Variability of agent costs from the base cost has a major impact on how agent types compare to one another. As the variability increases, the need for Best Fit and Co-opetitive agents becomes more evident. Compare Figure 6 to Figure 3. Figure 6 shows the individual profit ratio with agent costs having a standard deviation of 5. All previous figures have a deviation of 2.5.

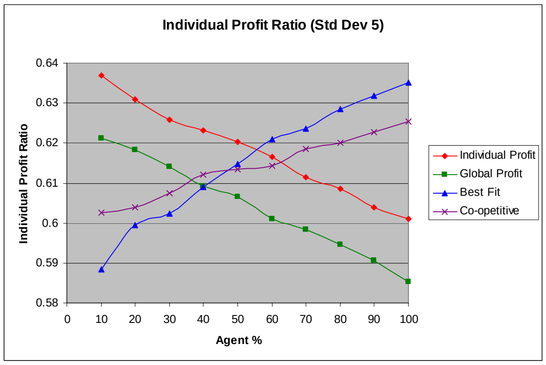


Figure 6: Individual Profit Ratio (Std dev 5)

The Best Fit and Co-opetitive agents cross the Individual Profit agent curve much earlier (55% and 65% respectively). They cross at 70% and 80% with a standard deviation of 2.5. The slope of the Co-opetitive agent increases and comes closer to that of the Best Fit agents.

Compare Figure 7 to Figure 4. At 100% of the same agent type, when the standard deviation is increased from 2.5 to 5, the difference between the number of tasks completed by the Best Fit and Global Profit agents increases from 3 to 6. The slope of the Co-opetitive agent curve increases and is much closer to the Best Fit agent.

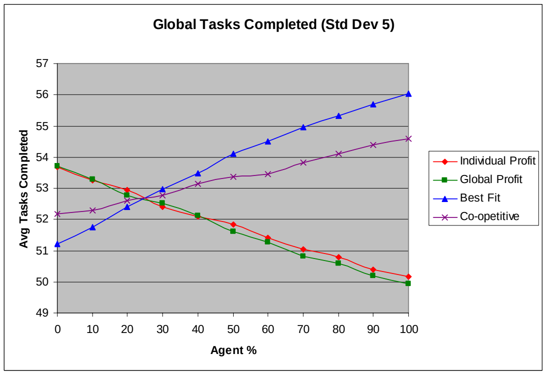


Figure 7: Global Tasks Completed (Std Dev 5)

Compare Figure 8 to Figure 5. At 100%, the spread between the Best Fit and Profit agents' profit ratio is increased from .025 (std dev 2.5) to .045 (std dev 5). The slope of the Co-opetitive agents' global profit ratio is increased and is much closer to that of the Best Fit agent.

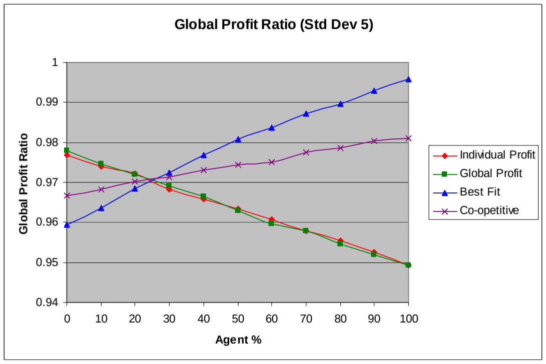


Figure 8: Global Profit (Std Dev of 5)

7 Future Work

Tasks could be made time sensitive. Kraus, Shehory, and Taase (2003) describe a discount factor that decreases the task pay over time. By applying a similar discount factor, agents could be encouraged to join coalitions more quickly. Agents could adjust their compromising ratio according to the discount factor of tasks.

Soh and Li (2004) evaluate their coalition forming strategy by measuring how difficult it is for agents to form coalitions. In our domain, this could be measured by the number of proposals made and how many eventually fail. This information could be used to improve agent decision making skills, so that fewer proposals fail.

8 References

Belmonte, M., Conejo, R., Perez-de-la-Cruz, J.L., Triguero, F., A Robust Deception-Free Coalition Formation Model, SAC '04, March 14-17, 2004, Nicosia, Cyprus

Blankenburg, B., Klusch, M, and Shehory, O, Fuzzy Kernel-Stable Coalitions Between Rational Agents. AAMAS'03, Melboure, Austrailia, July 14-18, 2003.

Blankenburg, B, Dash, R.K., Ramchurn, S.D., Klusch, M, Jennings, N.R., Trusted Kernel-Based Coalition Formation, AAMAS'05, July 25-29, 2005, Utrecht, Netherlands.

Brandenburger, A. M.; Nalebuff, B. J. Co-Opetition: the Game Theory That's Changing the Game of Business, Doubleday Publishing, New York, NY, U.S.A. 1996.

Caillou, P, Aknine, S, and Pinson, S., A Multi-Agent Method for Forming and Dynamic Restructuring of Pareto Optimal Coalitions, AAMAS'02, July 15-19, 2002, Bologna, Italy.

Dang, T., and Frankoviˇc, B., and Budinsk´av, I., Optimal Creation of Agent Coalitions for Manufacturing and Control. Computing and Informatics, Vol. 22, 2003, 1001–1031, V 2003-Jun-19

Gerber, A. and Klusch, M., Forming Dynamic Coalitions of Rational Agents by Use of the DCF-S Scheme. AAMAS '03, Melboure, Austrailia, July 14-18, 2003.

Klusch, M., and Shehory, O. A Polynomial Kernel-Oriented Coalition Formation Algorithm for Rational Information Agents, in Proceedings of ICMAS '96, 157-164, Kyoto Japan, 1996

Kraus, S., and Shehory, O., and Taase, G., Coalition Formation with Uncertain Heterogeneous Information. AAMAS '03, Melbourne, Austrailia, July 14-18, 2003.

Kraus, S., and Shehory, O., and Taase, G., The advantages of Compromising in Coalition Formation with Incomplete Information. AAMAS, pp. 588-595, Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2 (AAMAS'04), 2004.

Lau, H., Lim, W., Multi-Agent Coalition Via Autonomous Price Negotiation in a Real-time Web Environment. Intelligent Agent Technology, 2003. IAT 2003. IEEE/WIC International Conference on 13-16 Oct. 2003 Page(s):580 – 583, 2003

Lau, H., Zhang, L., Task Allocation via Multi-Agent Coalition Formation: Taxonomy, Algorithms and Complexity. Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'03), 2003

Li, C., and Rajan, U., and Chawla, S., and Sycara, Katia. Mechanisms for Coalition Formation and Cost Sharing in an Electronic Marketplace. Proceedings of the 5th International Conference on Electronic commerce. Pittsburgh, Pennsylvania, 2003

Li, X., Soh, L., Learning How to Plan and Instantiate a Plan in Multi-Agent Coalition Formation. Intelligent Agent Technology, 2004. (IAT 2004). Proceedings IEEE/WIC/ACM International Conference on 2004 Page(s):479 – 482, 2004

Scully, T., and Madden, M. G., and Lyons, G., Coalition Calculation in a Dynamic Agent Environment. Proceedings of the 21st International Conference on Machine Learning. Banff, Canada, 2004.

Soh, L-K, and Li, X. Adaptive, Confidence-Based Multiagent Negotiation Strategy AAMAS'04, July 19-23 2004, New York, USA.

Adapting Playgrounds using Multi-Agent Systems

Alireza Derakhshan⋆, Frodi Hammer⋆, Henrik Hautop Lund⋆,
Yves Demazeau†⋆ and Luigi Pagliarini‡⋆

⋆Maersk Mc-Kinney Moller Institute for Production Technology
University of Southern Denmark, Campusvej 55, DK-5230 Odense M, Denmark
kianosh@mip.sdu.dk, frodi@mip.sdu.dk, hhl@mip.sdu.dk

†CNRS, Laboratoire LEIBNIZ, Institut IMAG. 46, avenue Felix Viallet, F-38031 Grenoble Cedex, France
Yves.Demazeau@imag.fr

‡Accademia di Belle Arti di Roma
luigi@artificialia.com

Abstract

This paper introduces an approach on how versatile, dynamic and adaptive playgrounds can be developed using multi-agent systems (MAS), artificial intelligence (AI) and Playware technology. By modelling the children and the Playware playground into a MAS, knowledge about the children's favourite playing behaviour can be acquired. Experiments with children were conduced in order to record their playing behaviours on the Playware playground, which consists of tiles capable of sensing and actuation, and having abilities to communicate with neighbouring tiles. An ANN capable of classifying the children's behaviour within eleven categories (i.e. favourite playing behaviour) was trained using a subgroup of the children. Validating the ANN against the remaining children's behaviours, **93%** were correctly classified. An adaptive playground was implemented utilizing the ANN in classifying the children's behaviour real-time and thus allowing for the children's interest in the playground to be maintained and/or increased by using appropriate adaptation strategies.

1 Introduction

There exists an alarming tendency in the European population today with an significant increase of obesity related problems. The prevalence among children is as many as one in four affected in certain regions. Childhood obesity being an acute health crisis is a substantial risk factor and can lead to chronic non-communicable diseases. New approaches are needed to address the challenge of preventing obesity, particularly in the younger generations (Force and for the Study of Obesity (2002)).

A novel approach which encourages physical activity amongst children and youth is *Playware* (Lund and Jessen (2005), Lund et al. (2005)) which introduces technologies known from robotics, artificial intelligence and multimedia into play equipment. Playware is intelligent software and hardware which aims at producing play and playful experiences amongst its users. Furthermore, *Ambient Playware* is defined as Playware with ambient intelligence characteristics. Thus, Ambient Playware can be personalized, it is adaptive and it is anticipatory. This allows for an integration of Playware technology into the environment so that users can freely and interactively utilize it. We believe that future playgrounds can be made of several building blocks of Ambient Playware, which placed in a real physical environment, allows for the interactions with the building blocks to merge with the characteristics of the real world to letting creative and active plays emerge amongst its users.

Playware playgrounds contrast from today's computer games, which allow users to playing through virtual characters, by allowing the users to *be* the characters. However, the users can physically move and interact with and within the environment which differentiates Playware from virtual and augmented reality environments. By doing so, Playware playground merge the versatile, dynamic and adaptive behaviour of today's computer games with traditional physical playgrounds which ultimately can help in keeping the children's playing interest in the playgrounds and therefore proportionally increase their physical activity — i.e. the playground could adapt according to the children's behaviour.

2 Children, Playgrounds and Multi-Agent Systems

In order to develop versatile, dynamic and adaptive Playware playground knowledge about the children's favourite playing behaviour on the playground is necessary (Derakhshan and Hammer (2005)). This could be knowledge about favourite moving patterns, such as jumping, running, doing somersaults etc. Having this knowledge, it could be utilized in adapting the playground into favouring these behaviours[1].

Since all children are different their individual interest in the playground are also more or less different. Hence, in order to adapt the playground to maintain or even increase children's playing interests, the playground should be able to adapt accordingly. This implies that it must be possible to recognise and/or predict the behaviour of a child playing on the playground. However, when several children are playing together on the playground, the playground needs to be adapted in order to maintain several interests; thus, also the collective behaviour of the children needs to be recognised and/or predicted.

In order to recognise and/or predict children's interests the Playware playground is modelled in a multi-agent system. Using the VOWELS paradigm (Demazeau (1995)) MAS can be considered consisting of agents, environment, interactions and organisations. Thus, the Playware playground can be mapped into a MAS as shown in figure 1 where the children are considered as being agents, the playground being the environment, the interactions are those between the children and the playground (i.e. agent–environment interactions) or other children (i.e. agent–agent interactions) respectively. For now, only the agent–environment interactions will be considered and will be referred to as the agent's "situation". However, in future development, parts of the playground (e.g. the tiles) could become themselves agents in the system, and thus yielding for an extension of the agent–agent interactions in the system. Finally the two children playing together is the organization. Several organizations could exist if e.g. groups of children play independently of each other on the playground (Derakhshan et al. (2006b)). In this paper, the MAS is assumed only to consist of one organization.

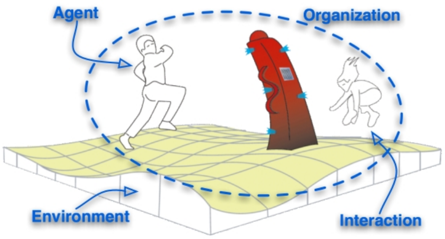


Figure 1: Mapping the Playware playground into a MAS.

2.1 Modelling the Children

Considering the children being the agents, the agents are supposed to be cognitive. That is, the children's behaviour, being a sequence of actions, is more or less intentional. Prior to performing an action, the child needs to deliberate according to its "playing intention" and its perception of the environment. However, the actions available to the child are limited to the agent–environment interactions of the Playware playground. In turn, these interactions define the possible behaviours of the agents which inhibit the system (Shoham et al. (2003)), i.e. the various behaviours children's.

Having several children playing together on the playground, collective behaviours occurs according to how the children organize themselves; that is, the organisation defines the role of the children. This agent–agent interaction originate from the children's "joint playing intention" on the playground. However, it is assumed in this paper, that the agent–agent interactions are given by the agent–environment interactions on the Playware playground.

2.2 Modelling the Environment

Before the agent–environment interactions can be defined, the environment needs to be modelled. The Playware playground consist of several building blocks, namely tangible tiles as illustrated in Figure 2 (Lund et al. (2005)), which allow for different playground morphologies according to the assembling. The assembled playground contribute to the play (e.g. by audio and/or visual interaction) allowing an interesting and dynamic storyline to emerge with which the children can merge with their own imaginary stories (Singer and Singer (1990)).

The tiles dimensions are $21cm \times 21cm \times 6cm$ (width, height, depth) and each includes: An Atmel ATmega128 microcontroller, a 4-way bidirectional communication bus for local communication

[1] E.g. if a child is playing the "smash game" (Headon and Curwen (2001)) on the playground, a location is indicated which the child has to "smash" by e.g. jumping on this location. If however the child playing the "smash game" favours running instead of jumping, the locations that the child has to smash should be adapted to be located farther apart increasing the use of the running behaviour in the game — i.e. the child's favourite behaviour.

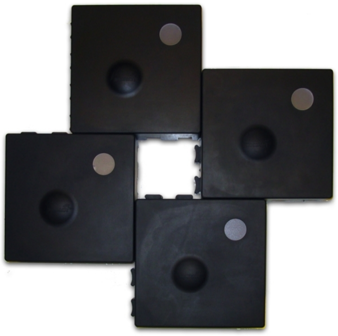


Figure 2: Four tangible tiles assembled into a playground. The light output is placed in the upper right corner of the tile while the "bump" indicates the position of the force sensor.

with neighbouring tiles at RS-232 level, four bright Light Emitting Diodes supporting output at different intensities of the colours red, green and blue and a Force Sensitive Resistor (FSR) sensor (FlexiForce A201) which can measure user interactions at dynamic levels.

Having the processing capabilities in each tile, and the local communication, the potential for each tile to become an agent is apparent.

2.3 Situating the Agents

Three physical parameters can be measured from the interactions with the tiles on the Playware playground, namely the force applied on a tile in a movement (p), the Euclidean distance travelled in a movement (d) and the duration of the movement (t) (Derakhshan et al. (2006b)). The three parameters are illustrated in Figure 3.

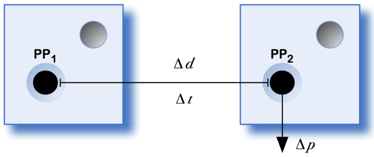


Figure 3: The three physical parameters that can be measured on the Playware playground — the movement travels from PP_1 to PP_2.

Combining these parameters the agent–environment interactions (i.e. the possible actions of a child) on the playground is defined. Having two states for each parameter, i.e. soft or hard force applied, near or far distance travelled and slow or fast duration of movement, eight actions are possible as listed in Table 1 and named according to their descriptive nature, e.g. a "walk" action has a soft pressure (contrary e.g. a "step" action), is at a near distance (contrary e.g. a "run" action) and is slow (contrary e.g. a "run" action). The states of the actions are defined according to appropriate predefined threshold values.

Table 1: The eight actions defined according to the force (p), distance (d) and duration (t) parameters.

Action	Δt	Δp	Δd
Walk	slow	soft	near
Touch	fast	soft	near
Step	slow	hard	near
Tug	fast	hard	near
Stretch	slow	soft	far
Stroke	fast	soft	far
Jump	slow	hard	far
Run	fast	hard	far

3 Adapting the Playground

Adaptation can be embedded on the Playware playground by considering the three phases as illustrated in Figure 4: *Observation* of the children's play, *Classification* of the children's behaviour and *Adaptation* of the playground. In the adaptation phase, an appropriate strategy must be chosen according to the children's interactions with the environment; thus, several different behaviours of the children need to be learned in order to apply the best adaptation strategy that promotes the activity level of the children.

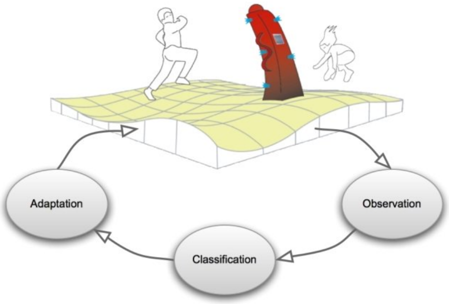


Figure 4: The adaptation cycle consisting of *observation*, *classification* and an appropriate *adaptation*.

3.1 Observing Children's Play

To get knowledge of children's behaviour on the Playware playground experiments were conducted. A

Playware playground, consisting of 36 tiles arranged in a 6 × 6 grid, capable of collecting data of the children's interactions was used together with video recordings. Prior to the experiments none of the children were informed about how to interact with the playground nor were they informed about the implemented game. By doing so, and by setting the experiments in a neutral environment as e.g. the children's kindergarten (see Figure 5), the experiments attempt to capture the blank intentions of the children.

For the experiments a game named "Bug Smasher" was implemented on the playground, which encourages the children playing to smash some bugs (indicated by the LEDs) which are wandering around the playground. The game has a number of internal states, interactions and feedback in accordance with the actions listed in Table 1.

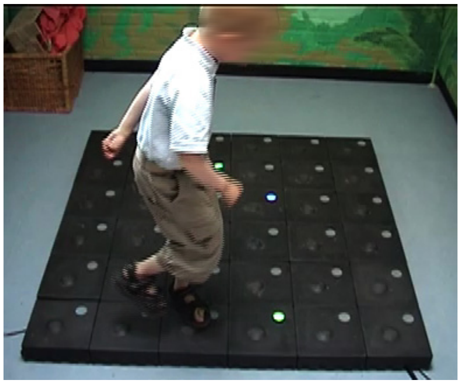


Figure 5: A 5 year old child playing a game on the Playware playground.

Two types of experiments were conducted; an *individual* experiment trying to settle the children's initial intention and a *group* experiment trying to settle the children's joint intention when playing together. In both experiments an external observer recorded the behavioural description of the children – i.e. classifying the cognitive agents (or, children) internal architecture according to their external description (Demazeau and Müller (1991)).

A total of 46 experiments were performed each lasted for 5 minutes in order to allow for the children's behaviour to evolve. All the children which participated were Danish and distributed among 27 males and 19 females. Two age groups were represented within the experiments; 37 were 5-6 years of age while 9 were 10-13 years of age.

In the individual experiments, three general observations were made from the children playing by the external observer (listed in Table 2): A *Playing* behaviour where the children interacted with the "Bug Smasher" game through the environment. A *Not Playing* behaviour where the children did not interact with the "Bug Smasher" game, but still interacted with the environment according to their intention. Finally, a *None* behaviour categorises the children who did not interact with the environment (i.e. their behaviour could not be measured by the playground). In total 17 children were *Playing*, 13 children were *Not Playing* and 8 children had a *None* behaviour.

Table 2: The general observation of the children's behaviour during the individual experiments.

Observation	5-6 years of age	10-13 years of age
Playing	10	7
Not Playing	13	0
None	6	2

In the group experiments, the children (only 5-6 years of age) were allowed to play two at the time on the playground. All the children participating had a *Playing* behaivour; however, two collaborative behaviours were noted by the external observed (listed in Table 3): A *Competing* behaviour where the children competed on squeezing the bugs in the environment. And *Cooperative* behaviour where the children cooperated on squeezing the bugs in the environment. However, the behaviour of the children would sometimes vary from being initially cooperative to occasionally competitive, e.g. some of the children pushed each other competing on squeezing a bug (see Figure 6). In total 3 groups were in general *Cooperating*, 1 group was in general *competing*, and 0 had a *None* collaborative behaviour as listed in Table 3.

Table 3: The general observation of the children's behaviour during the group experiments.

Observation	5-6 years of age
Cooperating	3
Competing	1
None	0

3.2 Classifying Children's Behaviour

The next phase in embedding adaptivity on the Playware playground is to classify the children's behaviour. The classes of behaviours can be defined according to the observations made by the external observer, excluding the *None* behaviours as these are unmeasurable from the playground. The behaviours are classified, as being from either an individual child or a group of children playing. If it is an individual child playing, it is classified within the two age

Figure 6: Two 5-6 years old girls competing on squeezing a bug in the group experiment.

groups (i.e. younger and older) and if the child is *Playing* or *Not Playing*. If a child is observed as *Playing* eagerly the class "Playing Fast" is used. On the other hand, if the child is barely *Playing* the class "Playing Slow" is used. If the child is *Not Playing*, but has a deliberate behaviour (i.e. the behaviour intentional) the class "Not Playing Continuously" is used. If the child is *Not Playing* in a chaotic manner (i.e. the behaviour is somewhat unintentional), the behaviour is classified as "Not Playing Discontinuously". Groups of children playing are classified according to if they are interacting with the game or not (i.e. *Playing* or *Not Playing*). If the children are *Playing*, they are next classified in accordance with their social behaviour; thus, "Playing Cooperating" and "Playing Competing" if the children are cooperaing or competing on squeezing the bugs respectively.

Thus in total, the eleven categories illustrated in Figure 7 were observed by the external observer, and can be used to classify the behaviour of the children on the Playware playground.

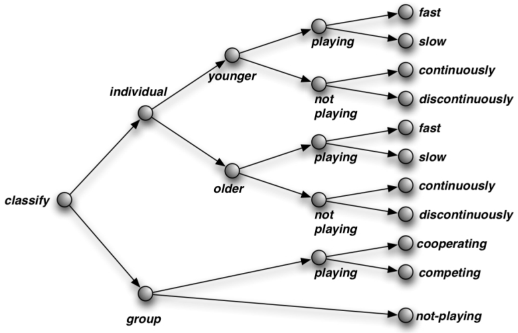


Figure 7: The eleven categories which classifies the children's behaviours on the Playware playground.

An artificial neural network (ANN) capable of classifying the children's behaviour according to eleven classes defined by the external observer has been designed. The architecture of the ANN is shown in Figure 8. The ANN inputs the child's perceptions of the environment and the corresponding actions performed by the child. The perceptions consists of the *direction* the child is moving on the playground, the pressed *tile's state* (i.e. the colour of the current tile) and the *neighbourhood state* (i.e. the colours of neighbouring tiles). The action input correspond to the three physical parameters (see Figure 3), that is; the *force* applied on the tile (p), the Euclidean *distance* travelled in the movement (d) and the *duration* (t) of the movement. To classify the behaviours defined in Figure 7, several perception/action inputs from the playground must be considered in sequence. Thus, history is added to the ANN as additional input neurons. History can be thought of as being the number of perceptions and corresponding actions which are necessary in order for the ANN to be able to classify a behaviour. A behaviour is found empirically to consists of 36 sequential actions (i.e. ensures the best convergence of the ANN) — which is analogue to the 36 tiles used in the experiments, and thus ensures that within the 36 steps, the child has the option to have an uniform distribution on the play area. The outputs of the ANN correspond to the eleven categories which classify the children's behaviour as illustrated in Figure 7. The ANN is implemented as a fully-connected three layered feed-forward network using the sigmoid activation function and trained by the back-propagation method.

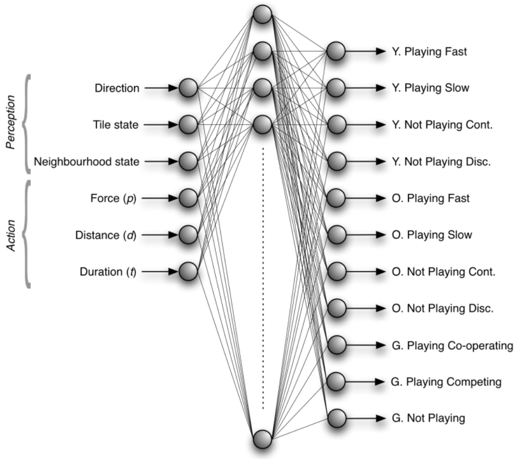


Figure 8: The architecture of the ANN design. For simplicity the history of the ANN is omitted in the figure.

The ANN was trained using a training set constructed from the observed behaviour of 7 children[2] as defining behaviours of the classes. For the defining behaviours 36 sequential actions were selected objectively by the external observer using video analysis. The configuration and convergence of the trained ANN are summarised in Table 4.

Table 4: Configuration and convergence parameter values of the trained ANN.

Parameter	Value
Number of input neurons	216
Number of hidden neurons	27
Number of output neurons	11
Learning rate, η	0.1
Training epochs	3000
Examples in training set	180
Training error, RMS	0.0034
Training error, MAX	0.020

The trained ANN was validated using a test set constructed from the collected data of the 31 remaining children from the experiments (excluding the children observed having a *None* behaviour) which were *not* used in the training process of the ANN. Before validating the ANN the subjects were categorised using an objective video analysis. The ANN classifies the validation subjects as listed in Table 5. It is apparent from Table 5 that 27 of the 29 subjects are correctly classified (93%); the misclassified cases was a child (S_{23}) categorized as "Younger Playing Fast" instead of "Older Playing Fast" and the two girls (S_{27}) which are categorized as "Group Cooperating" instead of "Group Competing".

When looking closer on the ANN output in Table 5 the classification percentage is varying. However, the output represents the overall classification of the children play during the experiment. Thus, these variations could be a result of the children changing behaviour during the experiments (see e.g. S_6). By investigating the moving average of the ANN classification for S_6 (see Figure 9) it is seen that the behaviour of S_6 is classified as either "Y. Not Playing Disc." or "Y. Not Playing Cont." during the experiment. To confirm that the changes in the ANN classifications are coherent with changes in the children's behaviour, a qualitative video analysis of the children's behaviour was made by a human classifier who classified the children's behaviour during sequences of 36 steps and assigning each classification with a confidence factor (i.e. illustrating the level of uncertainty). According to the human classifier S_6 is indeed changing her behaviour during the experiment (see Table 6). However, the behaviour changes noted by the human classifier are exactly those classified by the ANN in Figure 9 — it also appears that the confidence factor of the human classifier follows the magnitudes of the classifications by the ANN.

Table 5: ANN classification of the validation subjects. The subjects are numbered and marked as (Y)ounger/(O)lder and (M)ale/(F)emale. The ANN classification is shown as percentage of output, and bold is used to indicate the maximum output whereas underline indicates the expected classification of the subject according to the video analysis.

	Classification in percentage						
Subject(s)	Y. Playing Slow	Y. Not Playing Cont.	Y. Not Playing Disc.	O. Playing Fast	O. Playing Slow	Group Cooperating	Group Competing
$S_{1,YM}$	9	32	$\underline{\mathbf{55}}$	0	2	2	0
$S_{2,YF}$	6	10	$\underline{\mathbf{80}}$	2	2	0	0
$S_{3,YF}$	19	5	$\underline{\mathbf{63}}$	6	5	0	2
$S_{4,YF}$	1	$\underline{\mathbf{69}}$	30	0	0	0	0
$S_{5,YF}$	26	$\underline{\mathbf{50}}$	14	0	2	8	0
$S_{6,YF}$	1	$\underline{\mathbf{50}}$	41	0	5	0	3
$S_{7,YM}$	$\underline{\mathbf{42}}$	8	34	8	2	3	3
$S_{8,YF}$	12	25	$\underline{\mathbf{62}}$	0	1	0	0
$S_{9,YM}$	$\underline{\mathbf{55}}$	8	29	0	5	3	0
$S_{10,YM}$	$\underline{\mathbf{43}}$	14	39	2	0	1	1
$S_{11,YM}$	$\underline{\mathbf{72}}$	1	18	2	6	0	1
$S_{12,YF}$	3	$\underline{\mathbf{79}}$	7	0	1	10	0
$S_{13,YF}$	23	7	$\underline{\mathbf{56}}$	6	3	4	1
$S_{14,YM}$	$\underline{\mathbf{87}}$	1	3	9	0	0	0
$S_{15,YM}$	$\underline{\mathbf{68}}$	2	22	1	1	5	1
$S_{16,YM}$	$\underline{\mathbf{74}}$	0	1	15	1	9	0
$S_{17,YM}$	$\underline{\mathbf{59}}$	0	9	23	1	7	1
$S_{18,YF}$	13	8	$\underline{\mathbf{61}}$	8	1	1	8
$S_{19,YM}$	$\underline{\mathbf{41}}$	19	30	0	4	6	0
$S_{20,YM}$	$\underline{\mathbf{80}}$	0	0	16	1	3	0
$S_{21,OF}$	19	15	21	$\underline{\mathbf{43}}$	0	0	2
$S_{22,OM}$	12	7	10	$\underline{\mathbf{32}}$	16	7	16
$S_{23,OM}$	**33**	3	23	8	$\underline{17}$	9	7
$S_{24,OM}$	19	0	2	16	$\underline{\mathbf{31}}$	30	2
$S_{25,OM}$	7	1	3	$\underline{\mathbf{24}}$	20	22	23
$S_{26,YF\&F}$	14	1	1	5	3	$\underline{\mathbf{45}}$	31
$S_{27,YF\&F}$	15	1	1	10	4	**35**	$\underline{34}$
$S_{28,YM\&M}$	10	1	2	1	1	$\underline{\mathbf{60}}$	25
$S_{29,YM\&M}$	6	3	2	10	1	$\underline{\mathbf{41}}$	37

3.3 Adapting to Children's Behaviour

The final phase of Figure 4 is the adaptation phase. This phase includes extending the Playware playground into an Ambient Playware playground by uti-

[2] A complete training set could not be constructed, as only 7 of the 11 behaviours defined from Figure 7 existed in the data collected.

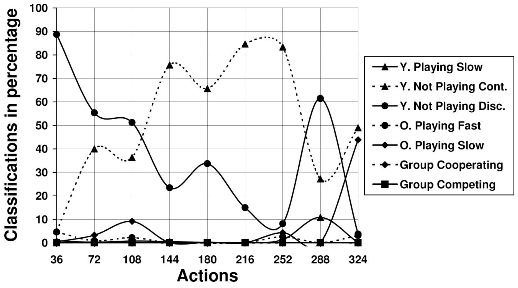


Figure 9: The moving average of the ANN classifications for the behaviour of S_6.

Table 6: Qualitative video analysis of the behaviour of S_6 by a human classifier. The confidence factors $\star$, $\star\star$ and $\star\star\star$ correspond to *vague observation*, *good observation* and *confident observation* of a behaviour respectively.

Actions	Human Classification	Confidence
0-36	Y. Not Playing Disc.	⋆⋆
37-72	Y. Not Playing Disc.	⋆⋆
73-108	Y. Not Playing Disc.	⋆
109-144	Y. Not Playing Cont.	⋆⋆
145-180	Y. Not Playing Cont.	⋆ ⋆ ⋆
181-216	Y. Not Playing Cont.	⋆ ⋆ ⋆
217-252	Y. Not Playing Cont.	⋆⋆
253-288	Y. Not Playing Disc.	⋆⋆
289-324	Y. Not Playing Cont.	⋆

lizing the classification capabilities of the ANN. By embedding the ANN into to playground, real-time adaptation of the environment according to the child's and/or children's behaviour can be realized. To do so, the implemented "Bug Smasher" game has been extending to include several adaptation strategies which adapts characteristics of the environment in order to encourage physical activity amongst the children. E.g. if a child is classified as "O. Playing Fast" the speed of the bugs is increased as is the distance to the child. Next, if the child slows down and gets classified as "O. Playing Slow" the bugs speed and distance to the child is decreased.

Preliminary experiments have showed, that the implemented adaptation strategies indeed encourages the children to a higher level of physical activity while still maintaining the interest in the playground — in some cases the interest even increased (Derakhshan et al. (2006a)).

4 Conclusions & Discussion

In this paper the novel Playware playground has been introduced and it has been shown how the playground can be adapted according to the behaviour of the children by the use of modern AI. The playground and the children playing on it have been modelled as a multi-agent system; modelling children as agents and the playground as the environment which allows for children–playground (agent–environment) and children–children (agent–agent) interactions and for the children as being organised. Having the MAS defined, an ANN has been trained using data recorded directly from the agent–environment interactions which allows for classifying the children's behaviour within eleven categories by 93%. The ANN has been embedded into the playground allowing for real-time adaptation with regards to the behaviour of the child playing. Adaptation strategies has been implemented *ad hoc* which allows for the playground to adapt in order to maintain and/or increase the child's interest.

Considering the importance and the significant increase of obesity related problems still little interdisciplinary work exists which encourage physical activity following the Playware approach by introducing technologies known from robotics, artificial intelligence and multimedia into play equipment.

However, the KidsRoom (Bobick (1996)) developed at MIT Media Laboratory offers a perceptually-based, multi-person, fully automated, interactive, narrative play space where children are taken on a fantasy journey through different worlds. Using a $24ft. \times 18ft.$ environment modelled as child's bedroom, the KidsRoom can track and analyse the behaviour of the children playing within the environment. Using sound, images, animations and music, the room can communicate with the children allowing the child (or children) to become a part of the story narrators, enabling them to change the course of the journey according to his/her behaviour and interaction with the room. Thus, the KidsRoom, like the Playware playground, allows for a high degree of freedom for child. The child does not have to know about the room's capabilities, i.e. he or she does not have to push certain buttons or do anything extraordinary for the storyline to proceed. However, the KidsRoom analyses the children's behaviour using computer vision constrained to certain fixed areas of the room, i.e. the child, in contradiction to the Playware playground, can only interact with the storyline at pre-defined areas in the room.

Also exergaming products such a *QMotions* and Konami's *Dance Dance Revolution* series of games are relating examples of entertainment media that mix physical activity with computer game playing. However, Playware offers the concept of building block game development which provides a much higher degree of freedom and flexibility in game designing. Thus, exergaming products can be considered as a sub-genre of Playware.

5 Current & Future Work

Currently work is in progress in two directions. First new experiments have been conducted to stress the effect of the adaptation on the children.

Experiments have been made where children are allowed to play the "Bug Smasher" game with and without the adaptation mechanism while measuring different entities such as response time on the playground, the number of interactions with the playground, the child's heart rate during play etc. to manifest the correlation between these different entities and adaptation. Preliminary results are indicating that the adaptation has the desired effect on increasing the physical activity of the children while playing.

Secondly new experiments are being made with a different type of game to settle the generality of the described approach taken in this paper.

In the future development of Playware playgrounds, emphasis should be placed on flexible user-centric Playware with enhanced ambient intelligence features which allows for further personalizability, adaptation and anticipatation according to the user's behaviour. This would indeed give the children and youth an exciting alternative to today's computer games and traditional physical playgrounds.

Acknowledgments

The authors would especially like to thank all the children who participated in the experiments and B. Iwersen from Hunderup Børnehave in Odense and H. "Malle" Malmros from Bryggens Børne- og Ungdomscenter in Copenhagen. Some of the hardware was developed by T. Klitbo and designed by C. Isaksen. KOMPAN A/S participated in the development of the tiles.

References

Aaron Bobick. The kidsroom: A perceptually-based interactive and immersive story environment. Technical Report 398, MIT Media Laboratory, Perceptual Computer Section, 1996.

Yves Demazeau. From interactions to collective behaviour in agent-based systems. In *Proceedings of the First European conference on cognitive science*, pages 117–132, Saint Malo, France, 1995.

Yves Demazeau and Jean-Pierre Müller. From reactive to intentional agents. In *Decentralized AI*, volume 2, pages 3–14. 1991.

Alireza Derakhshan and Frodi Hammer. "Adapting Playgrounds Using Multi-Agent Systems Based Upon Live Agents". Master's thesis, Maersk McKinney Moller Institute for Production Technology, University of Southern Denmark, Odense, Denmark, 2005.

Alireza Derakhshan, Frodi Hammer, and Henrik Hautop Lund. Adapting playgrounds for children's play using ambient playware. In *Proceedings of 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Beijing, China, October 2006a. (to appear).

Alireza Derakhshan, Frodi Hammer, Henrik Hautop Lund, and Yves Demazeau. Mapping children and playgrounds into multi-agent systems. In *Proceedings of 11th International Symposium on Artificial Life and Robotics*, Oita, Japan, January 2006b. ISAROB.

International Obesity Task Force and European Association for the Study of Obesity. Obesity in europe — the case for action, 2002. http://www.iotf.org/media/euobesity.pdf.

Robert Headon and Rupert Curwen. Ubiquitous Game Control. In *Ubicomp Workshop on Designing Ubiquitous Computing Games*, 30 September 2001.

Henrik Hautop Lund and Carsten Jessen. Playware – intelligent technology for children's play. Technical Report TR-2005-1, Maersk Institute, University of Southern Denmark, June 2005.

Henrik Hautop Lund, Thomas Klitbo, and Carsten Jessen. Playware technology for physically activating play. *Artificial Life and Robotics Journal*, 9 (4):165–174, 2005.

Yoav Shoham, Rob Powers, and Trond Grenager. Multi-agent reinforcement learning: A critical survey. Technical report, Stanford University, 2003.

Dorothy G. Singer and Jerome L. Singer. *The House of Make-Believe: children's play and the developing imagination*. Harvard University Press, Cambridge, MA, USA, 1990.

A Case-Based Learner for Poker

Arild Sandven* and Bjørnar Tessem†
Dept. of Information Science and Media Studies
University of Bergen, Norway
*arild.sandven@student.uib.no, †bjornar.tessem@uib.no

Abstract

Games have always been an important area for artificial intelligence research, and the topic has become increasingly more important during the later years due to the complexity of today's commercial computer games involving role playing adding psychological elements to the games. The poker card game constitutes a type of game where players have to handle not only statistical uncertainty, but also psychological aspects, which among humans usually are handled less by general statistical principles. Instead we handle psychological aspects by references to previously experienced situations. Thus, in this paper we will describe an approach to case-based reasoning as a technique to improve poker playing ability, as an alternative to rule-based and pure statistical approaches. Results show that the poker playing program is able to learn to play poker at the same level as rule-based systems even with a simple case-based reasoning approach.

1 Introduction

Games have always been a challenging domain for artificial intelligence (AI) research, from the start focusing on zero-sum, deterministic, perfect information games. Today computer games have turned into a big industry with an emphasis on role playing meaning that artificial intelligence components also has to handle psychological aspects of a game, observing behavior, interpreting, and suggest actions based on the interpretation of the situation [16].

The poker card game may on the surface seem to be a game almost of the classic type, but has important characteristics that also makes it a psychological game, involving both interpreting other players behavior and modifying own behavior as the game goes on. This is of course due to the lack of complete information inherit in the game, which makes the player dependent on probability assessments throughout the game. Any information available, including opponent behavior, will be used to assess the situation and decide play. Hence, unpredictable and confusing behavior is considered a winning strategy for poker players. For this reason we have seen several attempts to address poker playing as an AI research topic [3,4,6,9,11,13] many of them with a focus on opponent modeling and psychological aspects of poker [5,11].

It is accepted that humans handle many problem solving situations by referring to previous experiences in the form of cases or situations of similar type. Assessing situations in poker games is also of this kind, both regarding statistical and psychological factors involved. Within AI, Case-Based Reasoning (CBR) [12] is the technique used to perform this kind of problem solving. CBR has not to a great extent been used in game playing but has lately been taken up, for instance in [18] and also at an ICCBR workshop in 2005[1].

In this paper we will describe how a simple CBR tool (FreeCBR) may enable learning in the poker variant Texas Holdem, and we will document this by measuring the learning ability of a case-based poker playing program. We proceed the presentation by giving some background on the poker card game, poker playing software, and CBR in computer games in the next section, describe our *bot* (short for robot, term commonly used for artificial players in computer games) in section three, describe experiments and results in section four, and discuss the results in section five before we conclude.

2 Background

Important motivations for this research are the particular properties of poker which make the game an interesting game of study in AI, as well as the properties of CBR, which makes the approach ap-

[1] http://www.cs.indiana.edu/~davwils/iccbr05games/game-sim-ws.html

plicable to game playing in general, and in particular in poker.

2.1 The game of poker

Poker is a card game played among two and up to about ten players depending on the variant played. Texas Holdem, which is the variant used in our project, allows for 10 players. The goal of the game is to, based on the strength of your cards, outbet your opponents during several rounds of betting. When the last round of betting ends, the player with the strongest five card combination (pair, three of a kind, flush, straight, full house, and other combinations) wins the whole pot, which consists of the complete bets from all participating players. In each round players take turns betting, and are at any time free to withdraw from the game (fold), add an amount of money needed to equal the total highest bet so far in the game(call), or add more money than needed to equal the highest bet (raise). If all remaining players have called, the play continues to the next stage of play, which in Texas Holdem consists of showing more cards to the players. In Texas Holdem it is played four stages, the first with two private cards each, the next with added three common cards, the third with a fourth common card added, and the last with the fifth common card added. The combination for each player is based on the best five card combination of the seven available cards. A fourth kind of bet (check = a bet of 0) is allowed if nobody has bid anything in the stage so far.

High achievement in this game is of course dependent on ability to assess probabilities of winning given the available information in the cards dealt and the bets made so far. It has been developed highly sophisticated models for handling these probabilities given the known cards. However, as players we should also be informed by the bets given by other players, as these may indicate strong or weak hands on their part. This again opens for aggressive and passive playing styles, indicating willingness to bet more or less than the real value of any particular hand, and also bluffs, which are tactical bets on weak hands to make players with stronger hands than your own fold. Thus, achievement in poker also depends on ability to as often as possible to use psychological cues to correct our statistics based assessment of own hand strength compared to others.

In the particular version of Texas Holdem played by our software all raises have to be a fixed amount, and at any stage nobody should need to put in more than five times the raise, (i.e., only five raises are allowed in a round). In addition the game is played with *small blind* and *big blind* roles indicating that in the first stage the first player (small blind) is forced to bid one raise, and the second player (big blind) is forced to raise that bid.

2.2 Poker-playing software

Texas Hold'em has been the subject of extensive research from a group of researchers from the University of Alberta (UoA) over the past decade[2]. They have been able to create a program which is widely regarded as the strongest poker-AI in the world. Their prototypes, Loki and Poki have achieved great success against human opponents, and work is ongoing. They have developed and applied AI-techniques such as neural networks to be used by Loki and Poki. Also, a poker library, Meerkat, which contains useful poker tools, is made available to the public. As of late, the UoA research group have gone commercial, releasing their own Poker Academy™ (BioTools[3]). Lately, they have also widened their support to different types of poker, and they continue to support plug-in bots for amateur programmers.

In addition to the contributions from UoA and BioTools, several other researchers have explored poker from different angles. A simplified version of artificial poker and machine cognition was created as early as 1977 [9]. In the last decade, attempts have been made with poker based on Bayesian networks [13], evolutionary algorithms [3], and adaptive learning [11]. However, none of these have achieved the same attention or success as the UoA projects.

2.3 Case-based reasoning in computer games

Case-based reasoning [12] and its focus on single situational experiences as a problem solving tool provides a flexible and promising approach on domains with "*poorly understood problem areas with complex structured data that changes slowly with time and justification is required"* (Watson, [21]). The process of CBR is cyclical and consists of essentially four steps: retrieve, reuse, revise, and retain. Retrieve involves finding similar cases to a current case, reuse is to adapt and suggest a solution to the current situation, revise is to improve the solution based on other knowledge, and retain is to remember the current case and its solution by adding it to the case base. After each cycle the case base has been extended, and with even more knowledge hopefully has a better chance of providing good solutions to new similar problems. More learning oriented approaches to case-based reasoning is also denoted by the term case-based learning, also

[2] http://www.cs.ualberta.ca/~games

[3] http://www.poker-academy.com

often used synonymously with example-based learning or instance-based learning [14].

CBR does not have a long history for use in computer gaming. An early attempt was DeJong and Schulz's Othello playing software [7]. Later approaches include Fagan and Cunningham's work on prediction of human behavior playing Space Invaders [8], Powell et al. [17] who developed a CBR checkers player, Sánchez-Pelegrín and Díaz-Agudo [18] who describe the use of CBR on tactical decisions in a strategy game (C-evo), and Molineux et al. [15] and Aha et al. [2] who describe the use of CBR in different aspects of the real time strategy game Wargus.

3 The Case-Based Poker Player

The artificial player developed in this project was developed in Java and has the name Casey[4]. More precisely, Casey is a special type of client, which connects to a server and responds automatically to events. Contrary to normal poker clients, the human interaction and logic is replaced by a program which decides bids to be given in a game played on the server. Casey is implemented in accordance with the player interface developed at the University of Alberta. This ensures portability between bots to and from present and future poker frameworks. When Casey acts, he starts by sending the necessary game features to the CBR system. The CBR system then creates a target case, performs a search and returns the selected strategy based on Casey's preferences. The functionality of Casey's CBR system is described in the following.

3.1 The CBR-system

FreeCBR is a freeware CBR-system under the Public Domain-license. It offers interaction through a graphical user-interface, command-line, active-X or through Java libraries.

Although it is a quite simple system, FreeCBR proved sufficient for this project. However, research on different implementations of the CBR-system may be fruitful in order to maximize the results of Casey. If maximizing profit is the goal, FreeCBR may not be sufficient, but it is extensive enough to discover whether CBR has potential in this context.

With FreeCBR we may determine similarity using regular data types like String, MultiString, Float, Integer and Bool. The process of determining closest match can be found in the FreeCBR documentation[5]. The closest match is calculated using weighted Euclidian distance. First the case distance is computed from

$$d=\sqrt{\sum_{i=1}^{n} w_i \cdot d_i} \qquad (0)$$

where d_i and w_i are the measured distance and the weight for case feature i respectively. Then the similarity s between two cases is given by

$$s=100\cdot(1-d)\Big/\sqrt{\sum_{i=1}^{n} w_i} \qquad (0)$$

d_i is as mentioned the distance between the searched feature and the actual case feature. This value is a real number in the interval [0, 1] where 0 means exact hit and 1 means maximum distance. w_i is an integer >= 0, by default set to 5.

In addition, FreeCBR offers possibilities for more selecting alternative strategies for assessing similarity. In our solution we have used the default strategy, which in the documentation is named fuzzy linear and returns a similarity 1 if all features are equal or else the similarity given in equation 2.

3.2 Implementing the CBR-system

In Texas Holdem, one of the most important decisions is made once the player is dealt the first two cards. This stage is known as the *preflop* stage. The decisions of this stage are based on somewhat different features than in the other steges. Thus, we decided to separate the data into two case-bases, the *preflop* and *postflop* base.

Each case consists of two types of features. First, the indexed features are used for retrieval and should be predictive. In the context of poker we have chosen the following features

- *Hand strength (Hnd).* This is a relative numeric value calculated with the Meerkat Library. This is relative to the number of opponents in the hand. The values range from 0-100 where 100 indicates an unbeatable hand.
- *Relative Position (RPos).* A relative value to describe the players position. The value ranges from 0.0 to 1.0 where 1.0 means the player acts last, which is considered the best position.
- *Number of opponents (#Opp).* Indicates how many players are still in the hand. More precisely, *#Opp* is the sum of players who have already called a bet, and the players yet to act.
- *Bets to call(BTC).* If a player is in the big blind position and no-one has raised the pot, the value is 0, otherwise it would always be positive. If

[4] He should not be confused with CASEY (Kolodner, 1993).

[5] http://freecbr.sourceforge.net

the maximum number of raises is reached before the player has acted yet, the value may be 5.

All these features are highlighted by Sklansky and Malmuth [20], which is regarded as the leading authority on limit poker. If we recall the definition of predictive indices [12], they should be responsible for the way the case was solved, and influence the outcome. We argue that these features combined do just that. First of all, these features dictate how you play your hand because the end result is very much dependant on these features. Also, indices are abstract enough to be widened for future use. Since no specific rules or properties are modeled, these concepts apply to every kind of poker game, not only Texas Holdem. Finally, all indexes are numerical, and on an interval-scale, making them concrete and comparable.

Another type of features in the preflop casebase is the un-indexed features. They are not used in retrieval, but contain valuable contextual information. These indexes are

- *Strategy (Str).* Indicates which strategy is chosen for this scenario. Described in section 3.3.
- *Stage investment(SInv).* This is used to measure the possible negative consequences of a certain strategy. If a player has entered much money in the pot and lost, this will count as a very negative case.
- *Stage pot (Spot).* This is a measure of how much the other players put in the pot. Useful for deciding the most profitable strategy. If this value is high, and the player has won, this will count as a very positive case.
- *Result (Res).* The overall result of all actions in all stages combined. A positive number means the player won, otherwise, the game was lost.

These features can be used to maximize winnings, or minimize losses as they describe the success or failure of a specific case. Stage investment and stage pot describe the risk and reward factors of this particular stage, while result describes the net result of the complete game.

The postflop case base shares many of the features of the preflop case base. However, some differences exist. First of all, *Number of opponents* is no longer recorded. First of all, this was done because hand strength is reduced relative to the number of opponents. Secondly, Casey has information on previous raises (*bets to call*), giving a clear indication on whether the hand is good enough. Thirdly, and most importantly, with decent *hand strength* and several opponents, the pot odds (expected win divided by amount bid so far) increase with several opponents. A hand with no or little chance of winning will be folded regardless of the number of opponents, unless a bluff is likely to be successful.

Some features were also added. First of all, the potential features. The *positive potential* feature calculates the probability of improving the hand. This is very useful for strong *flush draws* and *straight draws*. The *negative potential* calculates the probability of getting outdrawn if the current hand is currently best. This means a high negative potential should induce betting and raising to make drawing opponents pay the maximum price to get lucky. Also, the *potsize* is included, as a big pot would make folding less attractive if you have any kind of *draw*.

Finally, the *relative position* feature is slightly altered. This feature now represents the position relative to the last person to act on this round (if someone bet in first position and was raised, he is now last to act). A raise immediately before your turn is not considered good. Below, we have summarized all features of the postflop case base.

- *Stage (Stg).* Indicates what stage this situation occurred in. 1 = Flop (Board cards 1-3) 2 = Turn (Board card 4) 3 = River (Board card 5)
- *Hand strength (Hnd).* As in preflop
- *Relative position (RPos).* A relative value to describe the players position. The value ranges from 0.0 to 1.0 where 1.0 means the player acts last, which is considered the best position. Note that if a player raises, the player before (if not folded) will be last to act, as he must close the betting.
- *Bets to call (BTC).* Indicates the number of bets and raises in front of the player. 0 means no-one has bet, 5 means betting is capped.
- *Positive potential (PPot).* Indicates how likely it is that the next card will improve your hand.
- *Negative potential (NPot).* Indicates how likely it is that you will lose, given that your hand is currently best.
- *Pot size ($Pot).* How big the pot is right now.

The postflop case base uses the same unindexed features as the preflop base.

After the features have been selected, reasonable weights for measuring overall distance needs to be made. Since we had no prior experience regarding these weights we chose default values wherever possible. This means that all indexed features were given the same weight. During initial testing, some experiments were made with differentiated weights, with especially priority on hand strength. This was later discarded since no immediate effect was observed.

3.3 Implementing strategies

In order to maximize profits in poker, players need to be unpredictable and tricky. To ensure that Casey exhibits these skills, a set of strategies has been developed. There are three available actions for the poker playing clients.

First, there is an aggressive action. This consists of BET/RAISE. More specifically, if nobody has bet, the action will be interpreted as a BET. If there is already a bet in the pot, an aggressive action constitutes a RAISE. Secondly, the passive action ensures that a player never folds, but does not enter more money into the pot than necessary to stay in. Passive actions consist of CHECK/CALL. A CHECK is performed if nobody has opened the betting, otherwise a CALL is performed. Finally, a FOLD action is available.

Note that check equals call, and bet equals raise when a player/bot performs an action. The current game state dictates which action is actually performed. All basic actions are summarized in table 1.

Action*	**Style**	**Description**
FOLD	SURRENDER	Fold
CHECK = CALL	PASSIVE	Check if no bet is made, call if bet is already made
BET= RAISE	AGGRESS-IVE	Bet if no bet is made, raise if bet is already made

Table 1: Player actions available.

To be able to perform advanced strategies, a *Strategy* feature was developed. We developed an algorithm deciding on a suitable strategy for any given game context using the case-bases. A *Strategy* consists of an initial action, and a follow-up response if applicable (depending on whether game-conditions permits or requires a second action from a player in a given stage). *Strategies* can also be categorized as honest and/or deceptive, and also as quit, steal or play. Table 2 gives an overview of strategies, or combinations of actions, that are implemented.

In playing we also used a random strategy option, particularly during training. For instance, if Casey is not yet familiar with a situation, a randomly selected non-fold strategy is chosen. This is done to build a decent sized case-base.

STRATEGY	TYPE	STYLE	GROUP	DESCRIPTION
FOLD	HONEST	SURRENDER	FOLD	Fold, check if no bet is made*
BLUFF	DECEPTIVE	AGGRESSIVE/SURRENDER	BLUFF	Bet/raise, fold if re-raised
CALL ANY	BOTH**	PASSIVE	PLAY	Check, call if bet is made
SEMI-BLUFF	DECEPTIVE	AGGRESSIVE/PASSIVE	PLAY	Bet/raise, call if re-raised
RAISE ANY	HONEST	AGGRESSIVE	PLAY	Bet/raise, re-raise if re-raised
CHECK RAISE	DECEPTIVE	PASSIVE/ AGGRESSIVE	PLAY	Check/call, re-raise again if re-raised

*If no bet is made, it is not smart to fold, a check will give more information at no cost. ** Context-dependent

Table 2: Strategies implemented

3.4 The decision-making process

In poker, several factors must be considered whenever an action is required. Great players balance all these factors, and choose an action to the best of their knowledge. The overall mantra, or heuristics, can be summarized into minimize your losses, but maximize your winnings. Casey uses the case-base when faced with a decision. Whenever a request for action is made from the dealer, a suitable strategy must be chosen based on the experiences recorded in the case base.

Initially, several options were considered. An optimistic best-match search was implemented and tested at first. This returned the strategy chosen from the closest matching case with a non-negative result. This resulted in several strange decisions as several cases were positive because of an incredible amount of luck (such as hitting two perfect cards in a row for a miracle straight). This was obviously not a winning play in the long run, and the algorithm was modified to take a larger number of similar cases into account before making a decision. This way, the effect of extraordinary results was minimized, and safer and more generally applicable strategies were favored instead.

In table 3 we see and example of a summary of previous similar experiences as presented to Casey. This is calculated from all closest matching cases within the given parameters, and is the basis of Casey's decisions. In other words, Casey is presented with a summary of the selected cases. We use the term *generalized case* for this summary, inspired by [1]. The table contains information of how often a game is won under similar circumstances, and the average results for each strategy.

The decision algorithm use information from the generalized case and mainly checks if the case base has enough similar cases, runs a random play if not, else checks whether there is a profitable strategy and plays this. If there is no profitable strategy, it checks whether this is a potential bluff situation and if so, runs a bluff.

STRATEGY	RESULT	FREQUENCY	WINS	LOSSES
Check-Fold	0	2	1	1
Bluff	-20	8	4	4
Call-Any	-35	1	0	1
Semi-Bluff	45	14	8	6
Raise-Any	120	15	7	8
Check-Raise	20	8	5	3
SUM Play	*150*	*38*	*20*	*18*
SUM Total	*130*	*48*	*25*	*23*

Table 3: The generalized case

Note that before we had enough cases matching the required similarity threshold we let Casey play a random strategy. This we did in order to fill the case base and get experience with different strategies in all situations. In the experiments, after 10.000 hands, the most common situations were familiar, and after about 20.000 hands random strategies were hardly used.

We also had to handle bluffs differently. We experienced that if a bluff was attempted and called, and it still led to a win for the bluffer, this would count as a good case, when in reality it meant that the bluff was unsuccessful. On the opposite side, if a bluff was randomly executed with great cards, this could lead to a fold if was re-raised (A bluff consists of BET-FOLD). We therefore decided to separate actions and strategies into three groups, PLAY, BLUFF or FOLD. The algorithm for selecting the best strategy from the generalized case thus only compares strategies in the PLAY group, and bluffs are chosen randomly only if no winning strategy is found.

After selecting a strategy according to these heuristics, the case is temporarily stored. At the end of each game the cases of that game is updated with information about the final result. The updated cases are then stored in the case base, and his knowledge base continues to grow.

4 Experiments and Results

The learning experiments for Casey consisted on runs of 50.000 poker hands. This took one week to run on this system, which of course limited the number of experiments to run. We ran he experiments with a similarity threshold of 95, which meant that if the case base did not have enough cases with higher similarity for the generalized case, Casey would choose a random strategy. Finally we used a maximum of 100 cases, i.e., the 100 most similar cases in a generalized case. Also decisions for case was based on the stage results rather than final results for the game, which is according to recommended poker playing behavior.

The first initial tests of Casey showed promise. He was run against a random player with absolutely no rules or learning abilities. As expected, Casey started winning almost immediately, needing only a few hundred hands to learn how to beat this opponent consistently. What he learned was to be aggressive on almost every hand, causing the random player to fold every 6th action (6 strategies to choose from), regardless of what hand he held.

For the main experiments, Casey played against instances of *RuleBot*, which is a not very sophisticated bot provided in the Meerkat library. However, the rules implemented suggest a safe and sound form of poker. It is not adaptive and without opponent modeling, but as opponent modeling is yet to be introduced in Casey, the bots compete on equal terms, treating each game as a game against unknown players.

The number of opponents should have an effect on your playing style. In a 2-man game, you figure to have the best hand every other hand, and should therefore play almost every other hand. In an 8-man game, such a strategy would probably be disastrous. We decided to test Casey against 3, 5 and 7 opponents to check his abilities with different types of games, and whether he was adaptable to such changes. Before changing the number of opponents the case base was reset, requiring Casey to start learning from scratch each time.

The success of Casey is dependant on two main factors. First, his results which are measured on a simple numeric scale. Second, his learning ability which is indicated by variation in results measured over time. Also, it is interesting whether he is able to mimic play related to other ideas, like the importance of position, or to protect your blinds. These abilities are not as important as the results and progress, but could give an indication of strengths and weaknesses by applying CBR in a game of poker. By investigating the database some of these questions may be answered, and we will return to this in the discussion.

Anyhow, two variables were measured and examined. First, the actual *result*, or how much Casey has won or lost totally at any time (balance would be an alternative concept). This gives an indication of his level of play. Secondly, we looked at the *fluctuations*, which is a measure of the difference between his highest and lowest result over a sequence of games. This indicates whether the results are produced by random luck (high fluctuation), but also it gives us an indication as to when his initial learning phase is over (when the fluctuations are fairly stable from phase to phase). We divided the 50.000 hands into 5 phases, each with 10.000 hands, and measured fluctuation within each of these phases.

Figures 1 and 2 give an overview of how results varied during the experiment runs and fluctuations in the five phases.

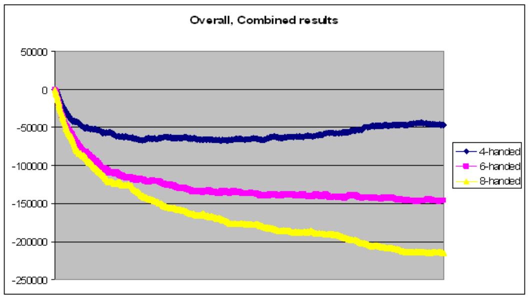


Figure 1: Overall results

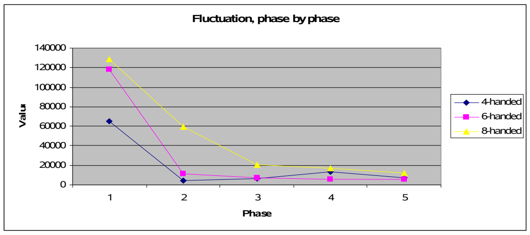


Figure 2: Summary of fluctuations

In the first phase, which spans from hand 0 to hand 10.000, Casey starts with absolutely no knowledge at all due to mainly random play, and to gain experience, he never folds. Naturally, playing every hand will lead to a negative result, as indicated in the figure above. As expected, the curve is steeper with 6 or 8 players as a wider variety of situations are encountered.

The second phase shows signs of promise in the games with fewer players. 4-handed, Casey already plays on level with RuleBot. 6-handed also shows significant improvement, both in terms of results and fluctuation, indicating that the initial learning phase is over. The 8-handed results are still far from satisfactory, but the curves indicate progress.

In the third phase, all simulation results seem to stabilize. Once more, Casey still is more successful in shorthanded (few opponents) games, showing a profit for the first time in the 4-handed game. 6-handed and 8-handed he is still losing, but once more showing progress from the previous phase. Fluctuations are still decreasing, an indication that he has not yet reached his peak.

In phase 4 Casey is still improving, but just slightly, and the results are getting more consistent. The 4-handed game is producing a nice profit, while 6-handed are almost on par, reproducing the indications from the previous phase. However, he is still trailing in the 8-handed game, which indicates that additional work is needed in order to produce a positive result here. It is also worth noting that the fluctuation of the 4-handed game has increased. This is a positive fluctuation, produced by winning play, and could indicate that Casey has bigger potential shorthanded, not yet reaching his peak. But he is still losing in the other games, and although not much, the fluctuations are not reduced significantly.

The last phase in the simulations presents some interesting results. In the 4-handed game, Casey still has a positive result. However, his average result is significantly reduced. If we look at the graph, this seems to stem from the last 3000 hands. This could indicate a natural run of bad cards and bad luck, or an especially lucky streak in the previous phase. Another possible explanation is that he has learned more about starting hands, being more selectively aggressive, and playing fewer hands. However, if we disregard the last 3000 hands, his average is on par with the results from previous phase (steady progress from 30.000 to 47.000), nurturing the assumption that he has experienced bad cards and bad luck. In the 6-handed game, the results are essentially the same. This indicates that the CBR system has taken Casey as far as it can. The fluctuations also stay the same, suggesting that further simulations would produce no significant progress or regress. Still, he plays almost on par with RuleBot, quite an accomplishment in our opinion. In the 8-handed game, fluctuation is once again reduced. Clearer indications would require at least 50.000 additional hands, as he probably has not reached his peak quite yet. His results are also still improving. However, since he seems to have reached his peak in the 4-handed game, we believe that additional test for the 8-handed game would only produce results approaching that of the 6-handed game.

5 Discussion

Casey did learn during experiments, but was unable to become a winning player against many opponents. However, playing fewer opponents, he performs very well. All-in-all, he performs reasonably well compared to the rule-based bots. We have seen that Casey clearly performs better when faced with fewer opponents. The probable explanation for this is his aggressive style during training, which is essential in shorthanded play [10, 19]. However, when opponents are many, much better cards are needed to win. Although this is picked up during play, the result seems to be that Casey himself becomes timid, folding marginal and great drawing hands. In other words, Casey learns hand strength to some degree, but not hand potential. On the other side, Casey seems to be able to exploit opponents betting patterns, by betting on weakness and folding when opponents show strength. Below we discuss some abilities that are considered desirable for good play and our assessment of Casey's performance regarding those abilities by using the plays database.

Starting hand requirements – Does he play the right hands?
One of the most important qualities of a winning player is playing the starting hands better than your opponents. Since good starting hands end up as winners more often, this gives the player a mathematical advantage. A good indication of this judgement is given by the percentage number of hands won when a hand is played. If we consider the 8-hand table, we can see that he wins only 6337 out of 15578 hands played in the flop stage, or about 40%. This indicates that his starting hands requirements need more work, as he clearly plays to many hands with little chance of winning.

Adaptability - Does he adapt to different number of opponents?
If we take a look at the percentage number of hands played in the three different simulations, we discover that his percentage drops with increased opposition. 4-handed, he plays about 37% of the hands. This percentage stays at 37% at the 6-handed table, partly because of the prolonged learning period necessary when facing more opponents, and therefore a broader variety of situations. However, 8-handed, his participation is down to 31%, with an even longer learning period. So, Casey experiences that it is difficult to win against more opponents, and plays fewer hands.

Pot odds concept – Can he identify valuable plays even with weaker hands?
From observing Casey, this does not seem to be the case. He throws away strong drawing hands from time to time, and occasionally also folds with *overpairs* (pair on hand with card value higher than any of the common cards). These problems are not easily corrected without implementing rules for pot odds and potential.

Bluffing – Is he able to exploit signs of weakness in his opponents?
Casey clearly exploits opponent weakness. he is aggressive in 4-handed games, and reduce his aggressiveness as the opposition increases. This is not merely a sign of pure bluffing skills, but also another indication that he adopts suitable playing styles according to changing table conditions.

Positional sense – Does he play more hands in late position?
If we look at the relative number of hands played in first vs. last position, Casey clearly prefers the latter. In the 8-handed game, Casey played 872 out of 6264 possible hands preflop from early position, or about 13% (first to act, position 2). When on the button (last to act, position 7), his participation was increased to 2187 out of 6203, or about 35% of the possible games. Playing more hands in late positions are considered good practice.

Blind play – Does he protect his blinds?
Having already highlighted the importance of positioning and exploiting weakness, it is also important to stop your opponents from doing the same to you. This is especially common against the blinds, as they have made forced bets and are likely to hold bad or mediocre hands. That means that it is necessary to protect your blinds by playing an increased number of hands that you usually would fold. Casey plays more blind hands than any of his opponents. Still, it is not easy to see whether Casey protects his blinds well or even over-protects them, making this a losing play. It could also indicate too low starting hand requirements, which is perhaps the most common source of losing.

Showdown – Does he know when to let bad hands go?
One last and perhaps most important ability is to let your bad hands go at the earliest possible moment. This is often measured by the percentage of *showdowns* (the deciding moment of a play when all the remaining players show their cards to decide who wins the pot) won. While showing down the best hand makes you money, it also creates an impression that you only play strong hands, setting up profitable bluffing opportunities later on. On the 8-hand table Casey wins, or breaks even 6337 times

out of 7488 showdowns, or about 85%. This is a strong result, and indicates an ability to throw away losing cards.

6 Conclusion

In short, our experiments indicate that a CBR system for poker play is able to develop playing strength and some recommended styles of play. Still, we see that some abilities are not learned satisfactory, partly due to lack of opponent models. Good opponent models would be able to quickly categorize opponents and use this information as an additional feature when selecting cases. In fact, CBR may indeed be of most value in opponent modelling. In addition we see a weakness in Casey's ability to consider pot odds and in assessing the start hands. Both opponent modelling, pot odds, and start hand assessment may be the focus of future work.

References

1. Aamodt, A. and E. Plaza (1994). "Case-based reasoning: foundational issues, methodological variations, and system approaches." AI Commun. **7**(1): 39-59.
2. Aha, D.W., Molineaux, M., and Ponsen, M. (2005). Learning to win: case-based plan selection in a real-time strategy game. Proceedings of the Sixth International Conference on Case-Based Reasoning (pp. 15-20). Chicago, IL: Springer.
3. Barone L. and While L. An Adaptive Learning Model for Simplified Poker Using Evolutionary Algorithms. In proceedings of Congress of Evolutionary Computation 1999 (CEC'99), July 6-9, Washington DC, pp 153-160, 1999.
4. Billings, D., Papp, D., Schaeffer, J., and Szafron, D. 1998. Poker as Testbed for AI Research. In *Proceedings of the 12th Biennial Conference of the Canadian Society For Computational Studies of intelligence on Advances in Artificial intelligence* (June 18 - 20, 1998). R. E. Mercer and E. Neufeld, Eds. Lecture Notes In Computer Science, vol. 1418. Springer-Verlag, London, 228-238.
5. Billings, D., Papp, D., Schaeffer, J., and Szafron, D. 1998. Opponent modeling in poker. In *Proceedings of the Fifteenth National/Tenth Conference on Artificial intelligence/innovative Applications of Artificial intelligence* (Madison, Wisconsin, United States). American Association for Artificial Intelligence, Menlo Park, CA, 493-499.
6. Billings, D., Davidson, A., Schaeffer, J., and Szafron, D. 2002. The challenge of poker. *Artif. Intell.* 134, 1-2 (Jan. 2002), 201-240.
7. De Jong, K., & Schultz, A. C. (1988). Using experience-based learning in game playing. Proceedings of the Fifth International Conference on Machine Learning (pp. 284-290). Ann Arbor, MI: Morgan Kaufmann.
8. Fagan, M., & Cunningham, P. (2003). Case-based plan recognition in computer games. Proceedings of the Fifth ICCBR (pp. 161-170). Trondheim, Norway: Springer.
9. Nicholas V. Findler. Studies in machine cognition using the game of poker. Communications of the ACM, 20(4):230--245, April 1977.
10. Jones, L. (2000). Winning low-limit hold'em. ConJelCo. Pittsburgh, Pa.,
11. Kendall, G. and Willdig, M. 2001. An Investigation of an Adaptive Poker Player. In *Proceedings of the 14th Australian Joint Conference on Artificial intelligence: Advances in Artificial intelligence* (December 10 - 14, 2001). M. Stumptner, D. Corbett, and M. J. Brooks, Eds. Lecture Notes In Computer Science, vol. 2256. Springer-Verlag, London, 189-200.
12. Kolodner, J. 1993 *Case-Based Reasoning*. Morgan Kaufmann Publishers Inc.
13. Korb, K. B., Nicholson, A. E., and Jitnah, N. 1999. Bayesian poker. In Proc. 15th Conference on Uncertainty in Articial Intelligence (1999). Morgan Kaufman.
14. Mitchell, T. M. 1997 *Machine Learning*. 1st. McGraw-Hill Higher Education.
15. Molineaux, M., Aha, D.W., & Ponsen, M. (2005). Defeating novel opponents in a real-time strategy game. In D.W. Aha, H. Muñoz-Avila, & M. van Lent (Eds.) Reasoning Representation, and Learning in Computer Games: Papers from the IJCAI Workshop (Technical Report AIC-05-127). Washington, DC: Naval Research Laboratory, Navy Center for Applied Research in Artificial Intelligence.
16. Alexander Nareyek. Artificial Intelligence in Computer Games - State of the Art and Future Directions. ACM Queue 1(10), 58-65, 2004.
17. Powell, J.H., Hauff, B.M., & Hastings, J.D. (2004). Utilizing case-based reasoning and automatic case elicitation to develop a self-taught knowledgeable agent. In D. Fu & J. Orkin (Eds.) Challenges in Game Artificial Intelligence: Papers from the AAAI Workshop (Technical Report WS-04-04). San Jose, CA: AAAI Press.
18. Rubén Sánchez-Pelegrín and Belén Díaz-Agudo. An intelligent decision module based on CBR for C-evo. In David W. Aha, Héctor Muñoz-Avila, & Michael van Lent (Eds.) Proceedings of the 2005 IJCAI Workshop on Reasoning, Representation, and Learning in Computer Games (http://home.earthlink.net/~dwaha/research/meetings/ijcai05-rrlcgw), Edinburgh, Scotland, 31 July 2005, pp. 90-94.
19. Sklansky, D. (1997). Hold'em poker. Two Plus Two Pub. Hendersen, NV.
20. Sklansky, D. and M. Malmuth (2001). Hold'em poker for advanced players. Henderson, Nev.
21. Watson, I. 1998 Applying Case-Based Reasoning: Techniques for Enterprise Systems. Morgan Kaufmann Publishers Inc.

Can We Prevent Collusion in Multiplayer Online Games?

Jouni Smed Timo Knuutila Harri Hakonen

Department of Information Technology
FI-20014 University of Turku, Finland
{jouni.smed, timo.knuutila, harri.hakonen}@utu.fi

Abstract

Collusion is covert co-operation between participants of a game. It poses a serious problem to multiplayer games that do not allow the players to share knowledge or resources with other players. Fair play requires that players act independently to gain best possible individual result and are also able to analyse the action-reaction loop of the other players. In this paper, we discern two basic types of collusion – among the participants and among the players – and give a classification of different collusion types. Since collusion cannot be tackled using real-time methods, we propose how to counteract collusion using prevention and detection methods.

1 Introduction

Cheating in a computer game can be divided to technical exploitations (e.g. tampering with the network traffic or cracking the software) and rule violations (Smed and Hakonen, 2006, §10). Many games assume that the players are rivals and, therefore, the rules forbid collusion, where two or more opposing players co-operate covertly towards a common goal. Especially imperfect information games such as poker, where each player has access only to a limited amount of information, forbid collusion by sharing information. If the players are physically present, it is easier to detect any attempts of collusion (e.g. coughs, hand signals, or coded language), whereas the anonymity of multiplayer online games makes collusion detection a difficult problem.

The colluders can share knowledge or resources among themselves, which means that collusion can take many forms: Co-operating players can engage in soft play and refrain from attacking one another. A gang of players can ambush and rob other players in a role-playing game. A novice chess player can resort to an expert – human or computer program – to make better moves. Participants in a tournament can pre-arrange the outcome of their matches to eliminate other players. Players belonging to the same clan can send numerous and seemingly independent complaints to the administrator to get an innocent player banned. A friend participating as a spectator in a first-person shooter game can scout the arena and reveal the location of the enemies.

In this paper, we aim at recognizing different types of collusion. This enables us to discern what kind of methods could be used in preventing them. After reviewing the background, we go through examples of co-operation in *Age of Empires* and collusion in poker. This leads us to classify collusion types into three distinct categories and discuss how to counteract them. These countermeasures either try to prevent the collusion beforehand or detect it afterwards.

2 Background

The fundamental problem of collusion recognition is to discern *deliberate* collusion from mistakes, luck, and skill (or the lack of it). Therefore, collusion recognition has to observe the following factors:

- Learning curve: As a player learns to play the game, his tactics and strategies tend to get better.
- Familiar company: A player has often a better success when the game-partners are familiar to him.
- Sensible play: Each player tries to minimize the worst possible outcome of the game.
- Conflicting interest: Decisions and actions concerning multiple players contain often interests that can be, at the same time, mutually beneficial and conflicting depending on the perspective.

An advanced player usually considers and seeks for actions that provide predictability from his perspective. This means that he acts so that the number of possible outcomes of the whole game narrows down. To achieve predictability he can aim at a stable situation in the game, and, to that end, players commonly develop repeating manoeuvres and routines that occur in certain situations (e.g. rules of thumb providing good a tactic or strategy). Another way to achieve predictability is to collude so that a (local) stability exists among the colluders.

A sensible player normally avoids wasting his time in unfruitful or unimportant conflicts and prefers agreements to preserve resources for some other purpose. Prolonged conflict has intentional (or non-operational) significance, because the effort to win has a cost (e.g. time, attention or resources required).

If two equal rivals engage in a conflict, they may know the equality beforehand, which leads to a cautious play. On the other hand, if equality is known only after the conflict, rivalry can be followed by tacit collusion (e.g. truce) until the status quo changes. If two rivals are not equal (in strength, resource types or knowledge) and do not engage in a conflict, we can suspect that they have formed a covert coalition.

2.1 Detecting Collusion

We conjecture that the smaller the set of choices the players face, the harder it is to hide the evidence of collusion. In card games, for example, the set of possible actions in a turn is limited, which means that it is difficult to evade detectable behavioural patterns. Since collusion is, fundamentally, one behavioural pattern, it is possible to develop methods for detecting it afterwards. For example, Rabiner (1989) describes a simple method for recognizing the tossing of a fake coin using hidden Markov models. Ercole et al. (2002) discuss how to capture colluders in a multiple choice examination. Also, the very fact that online poker game sites exist and thrive gives a real-world support to that it is possible to root out collusion in certain types of games – or at least give a believable threat of getting caught.

As the degree of freedom (i.e. the range of actions to choose from) increases, collusion detection becomes a tedious task. Especially online role playing game sites acknowledge the problem but admit that automatic policing is difficult to realize reliably. For example, *World of Warcraft* does not allow two rivaling guilds to collude and tries to detect it by tracking unusual patterns (Blizzard Entertainment, 2006).

Reliable collusion detection requires the following (True Poker, 2006):

- We can detect opportunities in order to recognize the possibilities for gain. The opportunities can be allowed by the rules or they can result from an unjust action.
- We can form a player profile by using pattern recognition. The profile concretizes different behaviours such as risk-taking, impatience, and favourite actions.
- We can analyse statistically the actions to determine the player's reputation. The reputation scheme can be open so that the players can rank one another. For example, a similar system is in use in Amazon.com where users can rank the products and the reviews of products, in eBay where the sellers are ranked, and in Xbox Live where TrueSkill system matches the players (Perry, 2005).
- We can analyse the play history to have a ground for penalization. By profiling the game players behaviour over time we can observe noticeable changes from usual habits which can indicate collusion.

All this analysis requires that we have some reference points such as:

- Game AI modelling expert play: How to value the actions and what is the best outcome that can be achieved with fair play?
- Game AI playing randomly: What is the least purposeful result?
- Player categorization: What is the typical selection of actions for a player?
- Player comparison: Are there correlations between the players' actions?

The reference point helps us in seeking out evidence of suspicious actions related to individual players or groups of players over time. We must also evaluate how much the action contributes to operational, tactical, and strategic level decision-making (i.e. the meaning and intention of a decision), and how rational the decision is in these levels (i.e. its degree of utility).

2.2 Classifications

Collusion classification is based on the type of agreement that the colluders have. The first categorization observes the *level of agreement*:

(i) Express collusion: The colluders have an explicit hidden agreement due to covert communication.

(ii) Tacit collusion: The colluders have no agreement nor covert communication but act towards a mutually beneficial goal such as attacking against the best player (to prevent him from winning) or the worst player (to force him out of the game).

(iii) Semicollusion: Collusion is limited to certain decision types and otherwise the colluders compete against one another normally.

The second categorization is based on the *content of the agreement*, which can be:

(i) Concealed stance: The colluder adapts different play methods against co-colluders and other players. For example, the colluders can engage in soft play and refrain from offensive action against one another, or even lose deliberately in a tournament to enhance co-colluders' position.

(ii) Knowledge sharing: The colluder gets support for decision-making from an expert co-colluder, who is not necessarily taking part in the game as a player. For example, a veteran player can help a novice by hinting which weapon to use against a given enemy.

(iii) Information sharing: The colluder exchanges or trades information related to the current game situation. For example, the colluder can signal to the co-colluders the movements of enemy troops.

(iv) Resource sharing: The colluder receives, donates or trades game-related resources with the co-colluders. For example, the colluders can whip-saw to raise the costs in the in-game market to force out non-colluding players.

Collusion can also combine these aspects. For instance, players can agree to collude (i.e. express collusion) using both soft play (i.e. concealed stance) and signalling (i.e. information sharing).

3 Examples

Since collusion is covert co-operation, any form of co-operation is possible to use in collusion. In this section, we take first a look at a commercial computer game that allows the players to engage in co-operation in many different ways and levels. After that we review the work done on counteracting collusion in poker.

3.1 Co-operation in *Age of Empires*

To demonstrate the possible ways of co-operation let us use the widely-known real-time strategy game *Age of Empires II: The Age of Kings* (Ensemble Studios, 1999) as an example. The game comprises at maximum eight players, which battle against one another or form alliances. Since co-operation is not forbidden by the game rules, there is no collusion. Nevertheless, since the game provides the players with a rich set of tools for co-operation, it gives good examples of the types of co-operation (and collusion) that the players can engage in.

- Forming alliances: Players can form alliances which means that they cannot attack one another's troops. Alliance indicates that the players are fighting for a common goal, and it is a prerequisite for other forms of co-operation (e.g. sharing knowledge).
- Sharing knowledge: A player can research a technology called 'Cartography' which allows him to see everything that his alliances see in the game world.
- Donating resources: A player can donate a part of his resources to an allied player. For example, resource dumping can help the other player to develop quickly militarily, while the other focuses on gathering resources. Donation can be also done by repairing allied player's building or healing his troops.
- Sharing control: Two or more humans can control one player in the game.
- Providing intelligence: When a player is defeated or quits, the fog-of-war is removed and he can observe the whole game world in a spectator mode. Although he cannot affect the game world any more, he can communicate to the other players what he sees.

It should be noted that none of this is forbidden but the game actually encourages the players to cooperate.

3.2 Collusion in Poker

Mental poker, introduced by Shamir et al. (1981), is ordinary poker where the players play by exchanging messages (e.g. over the phone) instead of cards. Originally, the problem was ensuring a fair deal. Crépeau (1986, 1987) recognised first the problem of collusion in mental poker but did not provide any methods for solving it. Similarly, the later studies have focused on ensuring fair shuffling, dealing and betting and the problems caused by collusion have been brushed aside. For a review on mental poker protocols, see Castellà-Roca (2005).

Hall and Schneier (1997) study the dealing and betting in online casinos but omit collusion. Johansson et al. (2003) use poker as an example to demonstrate that there are no pre-emptive nor real-time countermeasures against collusion. The organizer of an online game can try to track down the players, but that alone is not enough because the player's physical identity (i.e. his game account) does not reflect who is actually playing the game. Another approach is to analyse the game data to find out if there are players who participate often in the same games and, over a long period, profit more than they should (Vallvè-Guionnet, 2005). However, this kind of profiling requires a sufficient amount of game data, and collusion can be detected only afterwards.

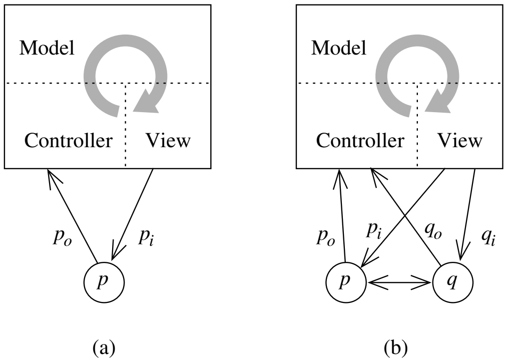
Poker players can collude in two ways: In active collusion, colluding players play more aggressively than they normally would (e.g. outbet non-colluding players). In passive collusion, colluding players play more cautiously than they normally would (e.g. only the one with the strongest hand continues while the others fold). Active collusion can be detected afterward by analysing the game data, but it is next to impossible to discern passive collusion from a cautious normal play (Johansson et al., 2003).

Online poker site terms and rules usually stipulate that anyone attempting to collude will be prohibit permanently using the services provided by the site and their account will be terminated immediately; for example, see Noble Poker (2006, §6.8) and Titan Poker (2006, §6.8). Collusion detection is mainly based on investigating complaints from other players, although some sites use methods for analysing the game data to find play patterns typical of collusion (True Poker, 2006).

4 Roles in Collusion

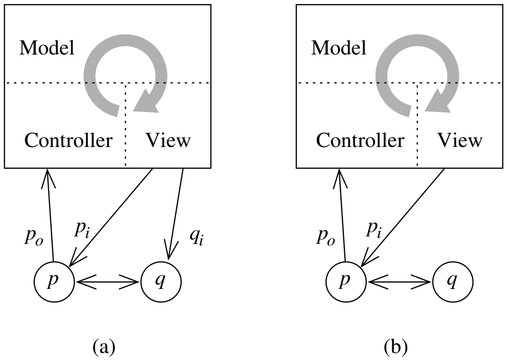
To discern different types of collusion, we must observe that the relationship between *players* and *participants* of the game is not one-to-one. For example, if two human participants take turns in controlling one game character, they appear in the game as one player. Similarly, a human can take part in the game in a spectator mode, where he is not a player but can participate by observing other players. In contrast, the player can also be a synthetic player, which is a computer-controlled entity that does not correspond to any human participant. Consequently, we must differentiate

(i) collusion among the players of the game, and
(ii) collusion among the participants of the game.



(a) (b)

Figure 1: Model–View–Controller and collusion. (a) Participant p has input p_i and output p_o, which him a player in the game. (b) When participants collude, they both can affect the game situation individually.



(a) (b)

Figure 2: (a) A participant has only a view to the game and can affect the game situation indirectly through a player. (b) A participant can communicate directly to a player.

To clarify the difference between these two types of collusion, let us look at the situation using Model–View–Controller (MVC) architectural pattern. The basic idea of MVC is that the representation of the underlying application properties (Model) should be separated from the way they are presented to the user (View) and from the way the user interacts with them (Controller); for a detailed discussion, see Smed and Hakonen (2006, §1.1). Figure 1(b) illustrates two colluders, p and q. If they both take part in the game as players, they both can affect the game individually (i.e. they both have access to the View and Controller part). In Figure 2, q is a non-playing participant, who has access only to the View part either directly (e.g. as a spectator) or indirectly (e.g. through p).

When players collude, they share and utilize knowledge or resources inside the game. This means that it is possible to detect the collusion, since the organizer of the game can analyse if the behaviour of players diverges from what can be reasonably expected. Conversely, detecting colluding participants is much harder, because it requires analysing the behaviour of individual players.

In the following subsections, we divide collusion types based on how the collusion works instead of what for it works. We describe collusion types, their indicators, and possible countermeasures.

4.1 Participant Identity Collusion

Let us first take a look at how a single player is perceived to participate in a game; see Figure 1(a). One of the simplest way to collude is to change or boost the participant that controls the player. Normally, we require that there is a unique human in the player loop. This concerns the relationship between a single player and the participant, and, thus, this collusion can be present in all other forms of collusion. Depending on how many parties are concerned, this kind of covert play can take the forms of player controller collusion or self-collusion.

4.1.1 Player Controller Collusion

When a player is not controlled only by a single human participant, the collusion takes the form of shared control. In the simplest form this can be achieved by replacing p by some other human participant (e.g. an expert player). A sweatshop (i.e. continuous play by taking turns) manufacturing digital game items or game avatars is an example this kind of collusion. To counteract a sweatshop, player's behaviour in a game session or in certain situations can be monitored and analysed in real-time (Furnell and Dowland, 2000). This kind of intrusion monitor techniques provide a way to verify the physical identity using the player account. Naturally, a simple analysis of the session logs can reveal if any of the participants has an exceptional metabolism.

Alternatively, the decision-making can be delegated so that p becomes a computer-controlled entity, a synthetic player. Game design can be used to make it harder for synthetic players to take part in the game. For this purpose, the game world should not have static or repeating elements, which can be utilized, for example, by scripting. Also, CAPTCHA methods (Golle and Ducheneaut, 2005) could provide a preventive way to tell apart human and synthetic players. The idea behind these methods is to have 'public' steganography for synthetic players that are easy for humans to solve.

In complex form, player controller collusion includes several parties (see Figure 2). For instance, the participant p can get help from a human helper q, who augments p's capabilities to play. Alternatively, the helper can be an automatized tool such as an auxiliary information tool, a decision aid, or an external simulator and analyser. The helper can also be the front-end for the participant such as a reflex augmentator or an action enhancer. These tools give the player abilities that he would not normally have. Such automatized tools can be detected by observing monotonic and repetitive action chains that do not take into account the nuances of the game situation.

4.1.2 Self-collusion

If a single participant controls multiple players (i.e. has alias players), he is engaging in self-collusion. This situation is illustrated in Figure 1(b), when $p = q$. Since it is easier for one participant to fake multiple players than multiple participants to pretend to be one player, the intrusion monitoring methods do not necessarily work in this case. Instead of preventing, we can model the player's decision-making process and measure deviations from it. However, this detection approach must account for the effect of the learning curve.

If we assume that there is no learning (i.e. the decision-making process is static), it is possible to apply traditional pattern recognition methods to the game data. For example, game state relationships can be modelled with self-organizing maps (Kohonen, 1995) and time series can be adapted using hidden Markov models (Rabiner, 1989). When the participant learns more about the game his decision-making process changes. To take this into account, the modelling phase makes use of typical learning curves and participant profiles. To ease the task further, the player types can be limited by offering dedicated servers for specific gaming types or target groups.

4.2 Inter-player Collusion

The second collusion type concerns multiple different participants as illustrated in Figure 1(b), when $p \neq q$. We can differentiate three forms of this collusion type: spectator collusion, assistant collusion, and association collusion.

4.2.1 Spectator Collusion

A spectating player, who participates in the game as a ghost, can reveal extra knowledge to another player. In other words, we have the situation illustrated in

Figure 2(a), where p_i is not congruent to q_i in terms of game playing.

Since the spectator mode can be used to gather intelligence, we can easily prevent it by providing the spectators a delayed feed of the game. Suppose the delay is one minute, which means that the spectators see the game events that have already happened and not the current situation. This reduces the value of the observations of dynamic information (e.g. the position of enemies). Naturally, the static information such as the number of enemies or the geography of the game world, has some value for the active players. Technically this approach means that the game server has to send two feeds – interactive live feed for the players and delayed feed for the spectators – which yields more network traffic. However, the protocol of the delayed feed can optimized with message aggregation and compression (Smed and Hakonen, 2006, §9.2). Also, the delayed feed can serve as an input for other desirable game functionalities, including recording, monitoring and verifying distributed control, and realizing public collusion detection.

4.2.2 Assistant Collusion

A player can be advised by another player who does not aim at winning but assist as a (possibly altruistic) sidekick. This situation is illustrated in Figure 2(a), when p_i is congruent to q_i, and q_0 is non-existent in practice. For example, if the team players in a first-person shooter can communicate verbally, a scout with binoculars can direct how the assaults should advance towards the enemy but do not otherwise engage in fighting. More extreme and direct example of this type of collusion is to join the enemy and act as a spy.

This kind of collusion takes form in miraculous escapes from situations that are caused by a failed tactic. Also, this kind of collusion can be used to reveal and neutralize opponent's tactical advantages.

Because the colluders act as players in the game, illicit assistance is hard to prevent. However, because the input feeds of the players are known, it is possible to deduce which player has the information that is possibly utilized by another player. Also, to test the suspected players, the game can set up sting operations or game playing traps by generating customized situations that are first observed by them.

Note that if q gets cut off from the game, he can continue assisting if p's input feed is forwarded to him, as in Figure 2(b). This forwarding can be as simple as sharing the same monitor display. In this case, collusion changes to player controller collusion.

4.2.3 Association Collusion

Active players can co-operate secretly due to shared interests or efforts, see Figure 1(b), where p_i is congruent to q_i. The difference to assistant collusion is that colluding players are in a symbiotic relationship and collusion provides them with a way to achieve their individual goals. Because cheating does not respect protocols or other rules, this form of collusion is almost impossible to prevent without a physical supervision of participants. In theory, it is also hard to detect it afterwards. However, collusion always produces patterns between the states and actions, and the simpler the game is, the harder it is to avoid leaving or hiding these traces. In other words, there is a limit when avoiding getting caught becomes more tedious than the game itself. Thus, if the collusion detection is not allowed to become a game in itself (e.g. by penalization), the situation is not so pessimistic in practice.

The association collusion can exist when a player carries out pre-emptive actions without any possibility to observe the tactical situation. For example, he can load his gun just in time, he can select a suitable weapon arsenal before the situation actually changes, or he can suddenly change direction to engage the opponents. The collusion produces also simultaneous actions and correlating tactics. For example, the players can specialize so that they complement each others.

The main difficulty of association collusion is that there is no way to prevent the ingenious use of external communication channels. Nevertheless, it is possible to reduce the benefit of these external channels from various aspects.

In the *game content* aspect, the first impression can be renewed by naming and labelling the game elements differently for each game instance. This idea is present in *Nethack* (DevTeam, 2006), where the player must use effort to re-learn the effect of the game items.

In the *player profile* aspect, the intentions can be conceptualized by incorporating them into the game GUI: The player must declare his stance towards the other players and make his tactical or strategic decisions explicitly predefined before the actions. The player profile have two immediate benefits: Firstly, the game play can be made smoother by making the actions context sensitive. Secondly, it gives name for each intention, which reference concepts for post-analysis programs.

In the *timing* aspect, the response window can be so short that the player has no time to discuss and analyse the situation with other players. Also, it is possible to bias the time-related phenomena within the limits of causality. For example, the game system can add jitter to communication to prevent use of timing covert channel (Yan, 2003).

In the *player identity* aspect, the player's identity can be obscured from the other players. The player can customize his avatar but the other players cannot see it modified. Also, the teams can be randomized, although this prevents co-operative play. This forces the colluding players to identity themselves to co-colludes inside the game world with an explicit signal.

In the *surveillance* aspect, if the colluding players are humans, the supervisors can also be humans. Actually, this introduces a higher level game between the colluders and the supervisors that some players may find more interesting. This calls for easy, public and open monitoring, where any participant can access the actions of some players. In this way the colluder do not know who supervises him and by what means and methods. The spectator feed is obvious mechanism to include this openness of decision-making and actions.

The transparency of actions must always be supplemented by substantial penalization (e.g. banning). To prevent collusion on penalization, there have to be a mechanism for the accused so that he can explain the suspicious actions later. It is worth noting that the required mechanisms already exist, since ordinary players commonly share experiences with friends and game community on online sites.

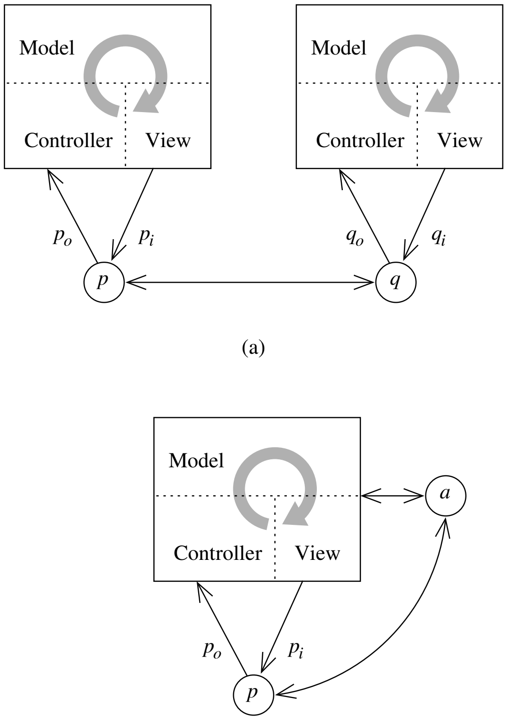
The surveillance can also include physical monitoring (web camera). This can be attached to reputation so that, for example, a player have to renew gained expertise level and reputation ranking by participating to supervised game instances from time to time (as in sports).

4.3 Game Instance Collusion

Collusion can be beneficial between the participants of different game instances, or between participants and administrators.

4.3.1 Multigame Collusion

The players of different game instances can also cooperate, as depicted in Figure 3(a). There are three typical intentions: Firstly, player q can provide intelligence to player p about the properties of the game world in general. For example, q can reveal secret game world locations or weaknesses of the synthetic players. Secondly, a group of players can scout in parallel different game servers to find a game instance that suits the group, and, after that, the whole group joins that game instance. Thirdly, especially when the game instances are not run in parallel, the players can collude on the tournament by fixing the match results together. Murdoch and Zieliński (2004) discuss more about collusion in tournaments.



(a)

(b)

Figure 3: (a) Collusion can span over multiple game instances. (b) A player can collude with a game administrator.

4.3.2 Insider Collusion

A player can be advised by the game administrator or the game developer. The situation is illustrated in Figure 3(b). This collusion can be very unintentional and tacit. For example, when a player calls for product support, the help desk can reveal accidently something on the game world.

5 Conclusion

In the title, we asked whether collusion can be prevented in multiplayer online games. To break down this question, we can now answer that it is possible – either by making it difficult to engage in collusion in the first place or by increasing the risk of getting caught afterwards. Consequently, the countermeasures are based on either prevention or detection. Their benefits are immediate: The players in multiplayer online games could be sure that there are no illicit alliances, gangs, clans or co-operation among rivals but everybody would have a fair playing field, which is one of the main goals of cheating prevention.

We also recognize that there is arising need for a third-party organization that grants and manages player licenses. Zetterström (2005) expresses a similar view and calls for an anti-cheating organization for online multiplayer games equivalent to the anti-doping organization WADA in sports. Such an organization could provide authentication of players and monitor their progress (e.g. maintain player rankings) but it would also provide a way to weed out players that systematically break the rules.

Acknowledgements
The authors thank Pyry Hakonen for pointing out collusion using parallel server scouting.

References

Blizzard Entertainment. Honor system Q&A with Kalgan. Web page, accessed April 28, 2006. ⟨http://forums.worldofwarcraft.com/thread.aspx?fn=blizzard-archive&t=24&p=1&tmp=1⟩.

J. Castellà-Roca. *Contributions to Mental Poker*. PhD thesis, Universitat Autònoma de Barcelona, Barcelona, Spain, 2005.

C. Crépeau. A secure poker protocol that minimizes the effect of player coalitions. In *Advances in Cryptology: Proceedings of Crypto '85*, volume 218 of *Lecture Notes in Computer Science*, pages 73–86. Springer-Verlag, 1986.

C. Crépeau. A zero-knowledge poker protocol that achieves confidentiality of the player's strategy or how to achieve an electronic poker face. In *Advances in Cryptology: Proceedings of Crypto '86*, volume 263 of *Lecture Notes in Computer Science*, pages 239–247. Springer-Verlag, 1987.

DevTeam. *NetHack 3.4.3*. 2006. ⟨http://www.nethack.org/⟩.

Ensemble Studios. *Age of Empires II: The Age of Kings*. Microsoft Games, 1999.

A. Ercole, K. D. Whittlestone, D. G. Melvin, and J. Rashbass. Collusion detection in multiple choice examinations. *Medical Education*, 36(2):166–172, 2002.

S. M. Furnell and P. S. Dowland. A conceptual architecture for real-time intrusion monitoring. *Information Management & Computer Security*, 8(2): 65–74, 2000.

P. Golle and N. Ducheneaut. Preventing bots from playing online games. *Computers in Entertainment*, 3(3):3–3, 2005. ⟨http://doi.acm.org/10.1145/1077246.1077255⟩.

C. Hall and B. Schneier. Remote electronic gambling. In *13th Annual Computer Security Applications Conference*, pages 227–230, San Diego, CA, USA, December 1997.

U. Johansson, C. Sönströd, and R. König. Cheating by sharing information—the doom of online poker? In *Proceedings of the 2nd International Conference on Application and Development of Computer Games*, pages 16–22, Hong Kong SAR, China, January 2003.

T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, Berlin, Germany, 1995.

S. J. Murdoch and P. Zieliński. Covert channels for collusion in online computer games. In *Information Hiding: 6th International Workshop*, volume 3200 of *Lecture Notes in Computer Science*, pages 355–369, Toronto, Canada, May 2004. Springer-Verlag.

Noble Poker. Terms and conditions. Web page, accessed April 27, 2006. ⟨http://www.noblepoker.com/termsofuse.html⟩.

D. C. Perry. Live in the next generation: The TrueSkill system, 2005. ⟨http://xbox360.ign.com/articles/662/662347p1.html⟩.

L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

A. Shamir, R. L. Rivest, and L. M. Adleman. Mental poker. In *The Mathematical Gardner*, pages 37–43. Prindle, Weber & Schmidt, Boston, MA, USA, 1981.

J. Smed and H. Hakonen. *Algorithms and Networking for Computer Games*. John Wiley & Sons, Chichester, UK, 2006.

Titan Poker. Terms and conditions. Web page, accessed April 27, 2006. ⟨http://www.titanpoker.com/termsofuse.html⟩.

True Poker. Anti-collusion, fraud detection, and random card shuffling. Web page, accessed April 26, 2006. ⟨http://www.truepoker.com/anti_collusion.html⟩.

C. Vallvè-Guionnet. Finding colluders in card games. In *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05)*, volume II, pages 774–775, Las Vegas, NV, USA, April 2005.

J. Yan. Security design in online games. In *Proceedings of the 19th Annual Computer Security Applications Conference (ACSAC'03)*, pages 286–297, Las Vegas, NV, USA, December 2003.

J. Zetterström. A legal analysis of cheating in online multiplayer games. Master's thesis, School of Economics and Commercial Law, Gothenburg, Sweden, 2005.

Mobile Games Pathfinding

Jarmo Kauko[*] Ville-Veikko Mattila[†]

[*]Nokia Research Center
Visiokatu 1, 33720, Tampere, Finland
jarmo.kauko@nokia.com

[†]Nokia Research Center
Visiokatu 1, 33720, Tampere, Finland
ville-veikko.mattila@nokia.com

Abstract

Pathfinding is a well-studied problem in computer games. The increasing complexity of games creates constantly new requirements for pathfinding systems. Meeting these requirements is especially hard for mobile devices due to their limited memory and processing power. This paper describes a system for fast and lightweight pathfinding in complex three-dimensional and dynamic environments, designed for mobile games. The system is based on visibility graphs and uses incremental heuristic searching. This paper also describes a novel way of combining these techniques. The experimental results on mobile devices show that the system is efficient and applicable for mobile games.

1 Introduction

In the recent years, the increased sophistication of graphics in video games has been yielding diminishing return in players' game experience. Stunning graphics is simply no longer the main driver for game sales, as the focus has been turning more towards gameplay and artificial intelligence (AI). AI is used in games to drive game characters, such as enemies and own troops that need to give an illusion of intelligent behavior; otherwise the game experience is ruined. Here, the objective of game AI is not to find an optimal solution in any given situation, but to make the game fun to play.

Games on mobile devices are currently using very primitive AI due to the devices' limited computation power. Thus, mobile games are often limited to simple shooting and racing games. Availability of AI could expand the range of games and genres that are feasible on mobile devices, considering, e.g., strategy games. In the long term, advanced AI could increase the popularity and acceptance of gaming among people, particularly adults, who are currently not playing games.

The development budgets and time of mobile games are typically strongly limited. Thus, game AI developers rarely have a chance to work on higher-level AI as much of the time is spent in struggling with low-level functionalities, e.g., in pathfinding. In the same time, retailers and network operators have raised the overall quality requirements of Java games, focusing on more complex 3D games. A clear qualitative improvement for AI techniques in mobile games will be dependent on appropriate interfaces to AI middleware to allow game AI developers to focus on higher-level creative AI tasks instead of worrying about low-level procedures. Naturally, such an interface should be generic enough to be applicable to a wide scope of mobile games. In Java games, an interface to an AI engine, running in the platform of mobile devices, could allow the adoption of computationally complex AI methods. However, this would require the standardization of an AI API in Java Community Process (JCP) to guarantee compatibility. Standard interfaces could also provide a foundation for the design of a dedicated hardware for game AI.

In this paper, we describe fundamental parts of our pathfinding system designed for mobile games. We start by introducing the underlying search space representation. We then continue with a short introduction to incremental heuristic search algorithms. After that, we describe the algorithm used in our pathfinding system. Finally, some experimental results evaluating the performance of the system are shown.

2 Visibility graphs

Visibility graphs are well studied and applied to various problems, such as urban planning and mobile robotics. For pathfinding, they can provide compact and accurate representations of two-dimensional surfaces containing polygonal obstacles. In order to use visibility graphs for our purposes, they had to be extended to operate on three-dimensional surfaces. Additionally, we wanted to be able to assign arbitrary cost factors to different parts of surfaces, instead of having only traversable and non-traversable parts.

Visibility graphs contain nodes at each convex corner of obstacles. Here, obstacles are not required to be non-traversable, but to have a higher cost factor than surrounding areas. The nodes that are visible to each other are connected. Due to the additional requirements, existing construction methods based on visibility tests cannot be applied to our visibility graphs. The traversability cost needs to be determined between every pair of nodes, resulting in $O(n^2)$ construction time in a graph containing n nodes. The traversability cost is computed by projecting the line between two nodes onto the surface and integrating the cost factors of the surface over the distance. The resulting graph can be used to find paths using shortest path algorithms such as the A* algorithm (Russel and Norvig 1995). However, since the start and end locations of a search can be anywhere on the surface, new start and end nodes needs to be generated. This requires computing edge costs between the new nodes and the existing nodes of the visibility graph. Since computing the edge costs is an expensive operation, the size of a graph dramatically decreases the performance of graph construction and searching. To overcome this problem, we allowed combining several smaller graphs to represent large environments. Hierarchical pathfinding uses similar approach to increase the searching performance at the expense of optimality (Botea, Müller and Schaeffer 2004), (Sturtevant and Buro 2005). Sub-graphs in hierarchical pathfinding are usually required to be connected, i.e., to have a valid path between each pair of vertices. In our system, sub-graphs are only used to determine sets of nodes that are potentially visible to each other. Edges between nodes in different graphs have to be defined explicitly.

2.1 Properties of an agent

One of the requirements for the pathfinding system was the ability to support various types of agents using a single graph. Using a separate graph for each type requires more memory, limits the variety of different agents, and requires more processing to adapt to dynamic changes in the game world. This was achieved by using cost vectors instead of scalar cost values. The components of the cost vectors present different types or sources of costs. Each agent has a property vector defining the influences of the components. Similarly, a movement constraint vector was used to describe strict limits, such as the maximum radius or a security access level.

A fundamental property in pathfinding is the size of an agent. Smaller agents should be able to use narrow passages and move closer to walls than larger agents. A traditional solution to handle the size of an agent is to use a free configuration space, i.e., expand the walls of the obstacles to match the shape of an agent (Lozano-Pérez and Wesley 1979). However, the structure of visibility graphs also provides interesting properties for supporting various sizes of agents using a single graph. The following example explains the intuition behind this (see Figure 1 a): Let E_{AB} be an edge between nodes N_A and N_B. Let O be a convex corner of an obstacle next to C_{AB}. Due to the presence of O, wide agents may not be able to use the edge E_{AB}. However, a node located to O, N_O, provides an alternative route through edges E_{AO} and E_{OB}. This way the visibility graph remains valid for different sizes of agents.

As the nodes are next to the obstacles, wide agents need to be displaced away from the walls. This was achieved by computing an offset vector for each corner. The offset vectors were also used to expand the geometry to compute the maximum radius of an agent for each edge. To estimate the maximum radius, only the corners that are visible from the end points of the edge need to be considered – if a corner is not visible, the corner that is blocking the visibility is closer to the edge. If an agent moves alongside an obstacle, at each point it needs to be displaced away from the obstacle by the radius of the agent. Thus, the path around a corner of an obstacle should be a smooth curve. In order to approximate this behavior, two nodes were used for sharp corners of obstacles. The both nodes have the same location but different offset vectors, as illustrated in Figure 1 b.

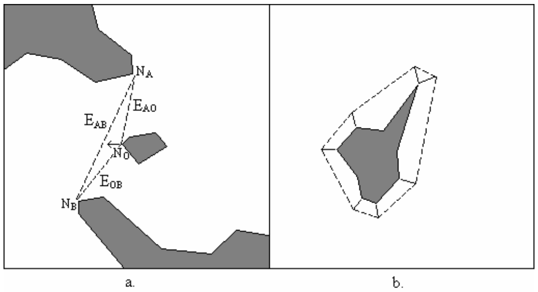


Figure 1: Examples demonstrating the construction of visibility graphs.

3 Incremental heuristic searching

Game worlds are often dynamic in a sense that they may change during the game play. These changes also occur, if an agent has incomplete or inaccurate knowledge of the game world and detects new information of it. Pathfinding needs to consider these changes and replan paths when necessary. Incremental search methods can provide fast replanning by using data from previous searches. They are successfully used, e.g., in robotics for navigation in partially known environments. However, they are not commonly used in games.

Games usually contain multiple moving agents and other highly dynamic factors. Replanning to adapt to these changes is infeasible, and usually not even reasonable, since the dynamic objects are likely to keep changing. Thus, games generally use local planning or steering algorithms to avoid these obstacles. Local obstacle avoidance requires an agent to deviate from the original path, which again is prone to the local minimum problem (Russell and Norvig 1995). Since returning to the original path may be troublesome, games often have to replan the whole path from scratch. However, incremental searching methods can be used to repair the original path whenever the agent drifts too far away from the path. This can reduce the frequent need for pathfinding substantially. Therefore, incremental searching is particularly important for pathfinding in games.

In order to use incremental searching, search data, including the priority queue and the cost values, has to be maintained for each search. Particularly, the required amount of memory in large grid-based graphs may be unacceptable for games. Since visibility graphs contain nodes only where a path turns, they are usually small and thus well suitable for incremental searching. However, using an incremental searching algorithm on a visibility graph is not straightforward, since the start and end nodes are generated. When agent moves along the path, the edge costs from the start node become invalid and need to be re-computed. As determining the edge costs is an expensive operation, replanning becomes very inefficient. The same problem applies to time-sliced pathfinding, where the searching process is divided over multiple frames, and the agent may move while searching (Higgins 2002).

A number of incremental algorithms exists for path replanning. The Focussed Dynamic A* (D*) (Stentz 1995) and D*Lite (Koenig & Likhachev 2002) algorithms are currently the most popular, because they efficiently use heuristics to determine, which parts of the path needs to be repaired. The method described in this paper is built on the D*Lite algorithm, but is applicable to any incremental heuristic search algorithm.

3.1 Lifelong planning A* and D*Lite

The D*Lite algorithm is based on Lifelong planning A* (LPA*) (Koenig, Likhachev, and Furcy 2004) that is an incremental version of the A* algorithm. LPA* reuses information from previous searches to adapt to dynamic changes in a graph. It also uses heuristics to focus the search and to determine which nodes need to be examined. If no information from previous searches is available, LPA* is similar to A* that breaks ties among nodes with the same f-value in favor of smaller g-values. Furthermore, it shares many interesting properties of the A* algorithm. It is relatively easy to understand, implement and extend.

Like A*, LPA* uses *g(s)* to indicate the cost from the start node and *h(s)* to approximate the cost to the end node for each node s. Since graph is allowed to change, *g(s)* is only an estimate of the start cost. To update the changed start costs, LPA* maintains *rhs(s)*, which is always one step ahead of *g(s)*. Formally,

$$rhs(s) = \begin{cases} 0 & \text{if s is the start node} \\ \min_{s' \in Pred(s)} (g(s') + c(s',s)) & \text{otherwise} \end{cases}$$

Here, Pred(s) is the set of predecessors of s and *c*(s',s*)* is the edge cost from s' to s. A node is said to be locally consistent, if its rhs-value equals to its g-value. Otherwise, the node is locally inconsistent. A node, s, is locally overconsistent if g(s) > rhs(s), and locally underconsistent if g(s) < rhs(s). LPA* maintains a priority queue U, containing all inconsistent nodes. U is sorted by a key, which consists of primary and secondary components:

$$k(s) = \begin{bmatrix} \min(g(s), rhs(s)) + h(s) \\ \min(g(s), rhs(s)) \end{bmatrix}$$

The LPA* algorithm starts by initializing the g-value of the start node to zero. All other g-values and rhs-values are initialized to infinity. Therefore, at the beginning, U contains only the start node. LPA* processes the nodes from U very similarly to how A* operates with the OPEN set. The algorithm makes the nodes locally consistent by removing the node with the smallest key from U and updating it and its successor nodes. Underconsistent nodes are processed slightly differently from overconsistent nodes. LPA* continues to process vertices until the goal node is locally consistent and has a smaller key than or the same key as the best node in U. If U becomes empty and the cost to the goal is infinity, there is no path available. The detailed description and analysis of the LPA* algorithm can be found in (Koenig, Likhachev, and Furcy 2004).

3.2 Our algorithm

The D*Lite algorithm uses heuristics to determine the searching order, i.e., the order, in which the nodes are processed. Our algorithm uses the same heuristics to determine when to compute the edge costs from the start node. As the D*Lite algorithm searches towards the start node, the edge costs from the start node are only needed to determine the total cost of a path. Before expanding a node, edge costs from the start node to all nodes with lower priority keys need to be determined. Otherwise, the nodes are not guaranteed to be processed in the right order,

```
procedure CalculateKey(s)
  01  return [min(g(s), rhs(s)) + h(start, s); min(g(s), rhs(s))];
procedure Initialize()
  02  U = V = CLOSED = Ø;
  03  for (all s ∈ S) rhs(s) = g(s) = ∞;
  04  rhs(goal) = 0;
  05  U.Insert(goal, [h(start, goal); 0]);
procedure UpdateVertex(u)
  06  if (g(u) ≠ rhs(u) AND u ∈ U) U.Update(u, CalculateKey(u));
  07  else if (g(u) ≠ rhs(u) AND u ∉ U) U.Insert(u, CalculateKey(u));
  08  else if (g(u) = rhs(u) AND u ∈ U) U.Remove(u);
procedure ComputeShortestPath()
  09  while (U.TopKey() < CalculateKey(start) OR rhs(start) > g(start))
  10      if (V.TopKey() < U.TopKey())
  11          u = V.Top();
  12          Compute the edge cost c(start, u)
  13      else
  14          u = U.Top();
  15          $k_{old}$ = U.TopKey();
  16          $k_{new}$ = CalculateKey(u);
  17          if ($k_{old}$ < $k_{new}$)
  18              U.Update(u, $k_{new}$);
  19          else if (g(u) > rhs(u))
  20              g(u) = rhs(u);
  21              U.Remove(u);
  22              Compute the edge cost c(start, u)
  23              for (all s ∈ Pred(u) ∪ {start})
  24                  if (s ≠ goal) rhs(s) = min(rhs(s), c(s, u) + g(u));
  25                  UpdateVertex(s);
  26              CLOSED.Insert(u);
  27          else
  28              $g_{old}$ = g(u);
  29              g(u) = ∞;
  30              for (all s ∈ Pred(u) ∪ {u})
  31                  if (rhs(s) = c(s, u) + $g_{old}$)
  32                      if (s ≠ goal) rhs(s) = $\min_{s' \in Succ(s)}$ (c(s, s') + g(s'));
  33                  UpdateVertex(s);
procedure UpdatePosition()
  34  if (start ∈ U) U.Remove(start);
  35  else if (start ∈ CLOSED) CLOSED.Remove(start);
  36  rhs(start) = g(start) = ∞;
  37  for (all s ∈ U) U.Update(s, CalculateKey(s));
  38  for (all s ∈ V) V.Remove(s);
  39  for (all s ∈ CLOSED)
  40      c(start, s) = ∞;
  41      V.Insert(s, CalculateKey(s));
procedure Main()
  42  Initialize();
  43  ComputeShortestPath();
  44  next = arg $\min_{s' \in Succ(start)}$ (c(start, s') + g(s'));
  45  while (not reached goal)
  46      if (g(start) = ∞) then there is no known path
  47      Move towards next
  48      if (reached next) next = arg $\min_{s' \in Succ(next)}$ (c(start, s') + g(s'));
  49      Scan graph for changed edge costs
  50      if (any edge costs changed)
  51          UpdatePosition();
  52          for (all directed edges (u, v) with changed edge costs)
  53              $c_{old}$ = c(u, v);
  54              Update the edge cost c(u, v);
  55              if ($c_{old}$ > c(u, v))
  56                  if (u ≠ goal) rhs(u) = min(rhs(u), c(u, v) + g(v));
  57              else if (rhs(u) = $c_{old}$ + g(v))
  58                  if (u ≠ goal) rhs(u) = $\min_{s' \in Succ(u)}$ (c(u, s') + g(s'));
  59              UpdateVertex(u);
  60          ComputeShortestPath();
  61          next = arg $\min_{s' \in Succ(start)}$ (c(start, s') + g(s'));
```

Figure 2: The algorithm for incremental searching in visibility graphs.

and the resulting path may not be optimal. If the agent's location has changed and the path needs to be replanned, edge costs from the new start node need to be computed to satisfy this condition. The first attempt was to compute edge costs to all processed nodes. However, if the agent has traveled a long way, this method is likely to compute more edge costs than what is necessary. If a valid path has been found, nodes with higher priority than the start node do not need to be considered. Therefore, the priority of the nodes can be directly used to determine, when to compute missing edge costs from the start node to the processed nodes.

The resulting algorithm is presented in Figure 2. The algorithm uses additional priority queue, V, to store all the nodes that potentially need to be updated for the current start node. It also uses CLOSED list to store all processed nodes, which is required for initializing V. Unlike D*Lite, our algorithm updates all h-values each time the agent moves in order to minimize the computation of the edge costs. This is acceptable since the number of nodes in visibility graphs is generally very low. This is done in UpdatePosition procedure (lines 34-41). Edge costs from the start node to nodes in V are computed in lines 10-12. The Main procedures of D*Lite and our algorithm are different, since our algorithm is applied to continuous environments. The algorithm was implemented to support time-sliced pathfinding, which was very straightforward as the algorithm supports agent movement during searching.

4 Experimental results

Two versions of the pathfinding system were implemented using Java and C. The implementations were algorithmically identical and no low-level optimizations were implemented. Performance of the implementations was compared to evaluate the benefits of having a pathfinding system for mobile games integrated to the platform. Two experiments were arranged to evaluate the system. The purpose of the first experiment was to measure the general performance of the system, compare the different implementations, and evaluate the scalability of the system to large graphs. The second experiment focused on algorithmic comparison of path replanning in visibility graphs.

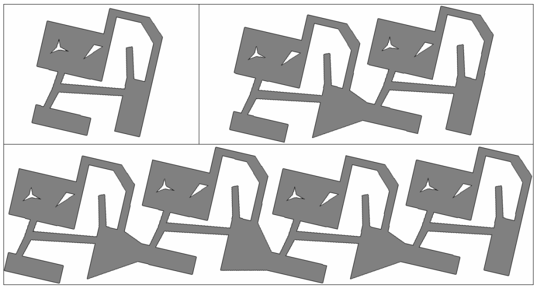


Figure 3: The test maps for general performance evaluation

4.1 General performance

The feasibility of the pathfinding system was evaluated by testing the performance of all critical operations. These operations include construction of a graph, dynamic modification of a graph and searching. The dynamic modification of a graph was allowed by adding arbitrary polygonal obstacles on the top of the movement surface. This requires generating new nodes to the convex corners of the obstacle, increasing the edge costs and updating the maximum agent size of nearby edges. Searching of a path was divided into two tests; generating the goal node and actual searching, which also determines edge costs from the start node. All of the operations were tested on three game maps, illustrated in Figure 3. Each map is represented using a single visibility graph to test the scalability without the usage of sub-graphs. The first map is relatively small, containing only 33 nodes. The second map is approximately two times as large as the first map, containing 69 nodes. Finally, the third map is approximately two times as large as the second map and contains 141 nodes.

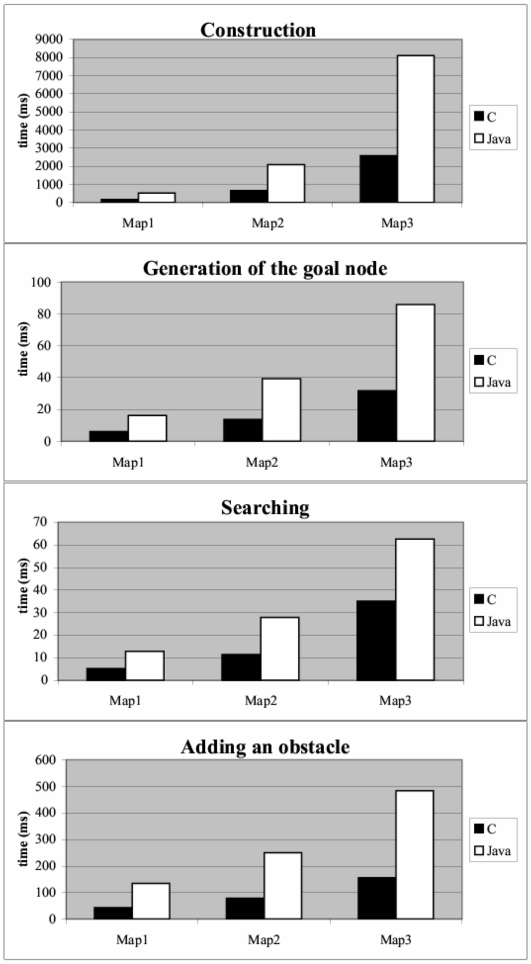


Figure 4: The results of the general performance evaluation.

The dynamic modification of a graph was tested by uniformly selecting ten locations from each map. A rectangular obstacle was added to each location. The searching was tested similarly by uniformly selecting ten locations from each map, and then finding a path between each pair of locations. This way the test contained more middle length paths, and less very short or long paths. The tests were performed on Nokia 6230i. The results of the tests are presented in Figure 4.

4.2 Path replanning test

Path replanning was tested to evaluate the benefits of incremental replanning in realistic in-game situations. For analytic and experimental results of the performance of the D*Lite algorithm, see (Koenig & Likhachev 02). As discussed before, the usage of local planning and steering algorithms often causes an agent to deviate from the path, therefore, requiring the whole path to be replanned from scratch. This experiment tests how different versions of our algorithm perform in these situations. The test map consists of three continents that are connected by two elevators as shown in Figure 5. The map is modeled using four visibility graphs; the main continent in the middle is divided into two graphs, and both side continents are modeled with one graph. The total number of nodes in the map is 68.

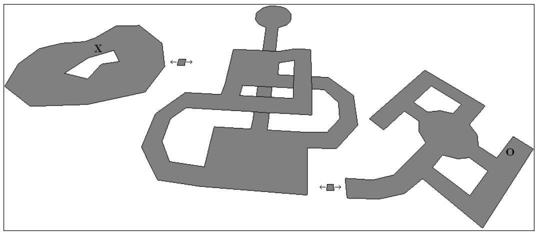


Figure 5: The test map for incremental searching tests. The map consists of three continents, connected by elevators.

Firstly, a path across the whole map was generated. In Figure 5, the start node of the path is marked with **o** at the right end of the map and the end node with **x** at the left end. Fifteen locations were selected along the path from the first three graphs. Each location was displaced away from the path and the path was replanned using the location as the new start node. The first algorithm replanned the whole path from scratch and computed the edge costs from the new start node to all nodes in that graph. The second algorithm replanned the new path from scratch as well, but used heuristics to determine the edge costs. The third algorithm was a basic incremental algorithm that computed edge costs from the new start node to all expanded nodes in the initial search. The final algorithm was the improved incremental algorithm as presented in Figure 2. The generation of the end node was excluded from the test, since the same end node can be used for recurrent searches. The test was performed on Nokia

6680 using the Java implementation. The results of the test are illustrated in Figure 6.

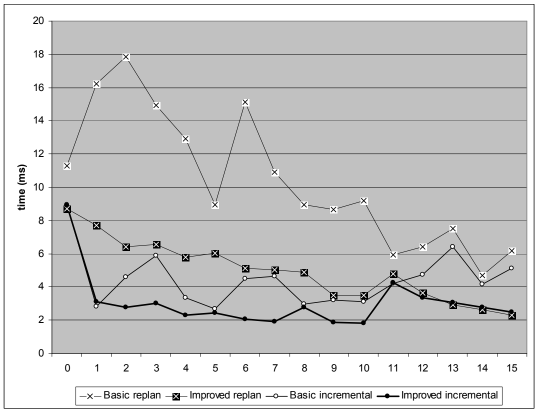


Figure 6: The results of path replanning test. The first column contains results of the initial search. The following columns (1-15) present the results of replanning the path from selected locations along the initial path.

5 Discussion

Visibility graphs allow representing complex three-dimensional environments using small number of nodes. However, determining edge costs between the nodes is computationally very expensive. As the results of the general performance evaluation indicate, the presented pathfinding system can handle small graphs relatively well in real time. On the other hand, very large graphs require much more processing time. However, a careful division into sub-graphs allows the system to scale well to large and complex environments. The system was also implemented to support time-sliced pathfinding, thus allowing multiple paths to be searched simultaneously without compromising the frame rate. The C implementation of the system proved to be remarkably more efficient than the Java implementation.

The incremental searching enables an efficient path replanning, also allowing global pathfinding and local steering algorithms to be combined more efficiently. As shown by the incremental searching test, the final algorithm replanned efficiently even for long paths. For very short paths, replanning from scratch can actually be slightly faster than incremental searching, since the incremental searching requires that the h-values are updated. In complex multi-agent environments, paths are often long and have to be replanned repetitively. Therefore, the usage of the incremental replanning algorithm can reduce the total searching load substantially.

One of the major bottlenecks was the math required for generating nodes and determining edge costs. Floating-point math was used to perform these calculations. Since the utilized mobile devices did not have a floating-point processing unit (FPU), the performance of floating-point math was a critical issue.

6 Conclusions

This paper described a pathfinding system designed particularly for mobile games. The main objective was to study the benefits of having such a solution integrated to a mobile platform to improve the performance of Java games. The system uses visibility graph-based approach to represent movement surfaces in two- or three-dimensional game worlds. It was designed to support various types of agents using a single graph, thus keeping the memory consumption low. This would also allow a fast adaptation to the dynamic changes in a game world, not limiting the variety of different agents. Large graphs can be divided into sub-graphs to guarantee scalability. As the experimental results show, the system was able to perform complex pathfinding tasks in real-time.

The pathfinding system uses incremental heuristic searching, which allows a fast path replanning. This is particularly important to combine global pathfinding and local obstacle avoidance behaviors. The paper presented a novel technique that allows incremental and time-sliced pathfinding on visibility graphs. The resulting algorithm was compared to less advanced replanning methods. The experimental results show that the presented algorithm allows fast path replanning regardless of the length of the path.

The C implementation of the system proved significantly more efficient than the corresponding Java implementation, indicating that Java games would remarkably benefit from AI solutions integrated to the platform. Obviously, this would also cut the development time and costs of Java games, as well as allow developers to concentrate on more advanced AI.

Acknowledgements

We would like to thank our colleagues in Nokia Research Center Multimedia Technologies laboratory for constructive discussion on the subject.

References

Botea, A., Müller,M., and Schaeffer, J. Near optimal hierarchical path-finding. *Journal of Game Development*, 1(1):7–28, 2004

Higgins, D. Pathfinding Design Architecture. *AI Game Programming Wisdom*, pp. 122-132, 2002.

Koenig, S., and Likhachev, M. D* Lite. *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, pp. 476-483, 2002.

Koenig, S., Likhachev, M., and Furcy, D. Lifelong Planning A*. *Artificial Intelligence* 155:93-146, 2004.

Lozano-Pérez, T., and Wesley, M. An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles. *Communications of the ACM* 22:560-570, 1979.

Russell, S., and Norvig, P. *Artificial Intelligence: A Modern Approach*, 1995.

Stentz, A. The Focussed D* Algorithm for Real-Time Replanning. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pp. 1652-1659, 1995.

Sturtevant, N., and Buro, M. Partial Pathfining Using Map Abstraction and Refinement. *Proceedings of AAAI*, pp. 47–52, 2005.

Game Theoretic Methods for Action Games

Ismo Puustinen Tomi A. Pasanen

Gamics Laboratory
Department of Computer Science
University of Helsinki

Abstract

Many popular computer games feature conflict between a human-controlled player character and multiple computer-controlled opponents. Computer games often overlook the fact that the cooperation between the computer controlled opponents needs not to be perfect: in fact, they seem more realistic if each of them pursues its own goals and the possible cooperation only emerges from this fact. Game theory is an established science studying cooperation and conflict between rational agents. We provide a way to use classic game theoretic methods to create a plausible group artificial intelligence (AI) for action games. We also present results concerning the feasibility of calculating the group action choice.

1 Introduction

Most action games have non-player characters (NPCs). They are computer-controlled agents which oppose or help the human-controlled player character (PC). Many popular action games include combat between the PC and a NPC group. Each NPC has an AI, which makes it act in the game setting in a more or less plausible way. The NPCs must appear intelligent, because the human player needs to retain the illusion of game world reality. If the NPCs act in a non-intelligent way, the human player's suspension of disbelief might break, which in turn makes the game less enjoyable. The challenge is twofold: how to make the NPCs act individually rationally and still make the NPC group dynamics plausible and efficient?

Even though the computer game AI has advanced much over the years, NPC AI is usually scripted (Rabin, 2003). Scripts are sequences of commands that the NPC executes in response to a game event. Scripted NPCs are static, meaning that they can react to dynamic events only in a limited way (Nareyek, 2000). Group AI for computer games presents additional problems to scripting, since the NPC group actions are difficult to predict. One way to make the NPCs coordinate their actions is to use roles, which are distributed among the NPCs (Tambe, 1997). However, this might not be the optimal solution, since it means only the distribution of scripted tasks to a set of NPCs.

Game theory studies strategic situations, in which agents make decisions that affect other agents (Dutta, 1999). Game theory assumes that each agent is rational and tries to maximize its own utility. A Nash equilibrium is a vector of action selection strategies, in which no agent can unilaterally change its strategy and get more utility. A Nash equilibrium does not mean that the agent group has maximal utility, or that it is functioning with most efficiency. It just means that each agent is acting rationally and is satisfied with the group decision.

If a NPC group can find a Nash equilibrium, two important requirements for computer game immersion are fulfilled: each NPC's actions appear individually reasonable and the group seems to act in a coordinated way. This requires that the NPC has a set of goals, which it tries to attain. The NPC gets most utility from the group actions which take it closer to its goals. The NPC's goals are encoded in a utility function. Defining the utility function is the creative part of utilizing game theory in computer game design, since Nash equilibria can be found algorithmically.

In our research we study creating a well-working utility function for a common class of computer action games. We also consider the problems of finding a suitable Nash equilibrium in reasonable time.

2 Game Theory

Game theory helps to model situations that involve several interdependent agents, which must each choose an action from a limited set of possible actions. Choosing the action is called playing a *strat-*

egy. One round of strategy coordination between agents is called a *game*. The game can be represented as a set of *game states*. A game state has an action vector with one action from each agent. Thus a game has k^n game states, if there are n players in the game with k possible actions each. Each agent gets a utility from each game state: the game states are often described as utility vectors in a matrix of possible agent actions. If an agent's strategy is to play a single action, it is called a *pure strategy*. Sometimes an agent gets a better expected utility by selecting an action randomly from a probability distribution over a set of actions; this is called a *mixed strategy*. The set of actions an agent plays with probability $x > 0$ is the agent's *support*.

As stated before, a strategy vector is a Nash equilibrium only if no agent can unilaterally change their action decision and get a better utility. Nash (1950) proved that every finite game has at least one Nash equilibrium. When a pure strategy Nash equilibrium cannot be found, a mixed strategy equilibrium has to exist. If we find a Nash equilibrium, we have a way for all participating agents to act rationally from both outside and group perspective.

Finding a Nash equilibrium from a game state search space is non-trivial. Its time complexity is not known (Papadimitriou and Roughgarden, 2005). The n-player time complexity is much worse than the 2-player case (McKelvey and McLennan, 1996). The problem is that the players' strategy choices are interdependent: whenever you change one variable, the utilities for each other player also change. The problem of finding all Nash equilibria is also much more difficult than the problem of finding a single Nash equilibrium.

How can we know if the Nash equilibrium we found is good enough? It also turns out to be quite difficult. Determining the existence of a Pareto-optimal equilibrium is NP-hard (Conitzer and Sandholm, 2003). Even finding out if more than one Nash equilibrium exists is NP-hard. However, some places in the search space are more likely to have a Nash equilibria, and that heuristic is used in a simple search algorithm to find a single Nash equilibrium (Porter et al., 2004).

The search algorithm is based on the heuristic that many games have an equilibrium within very small supports. Therefore the search through the search space should be started at support size 1, which means pure strategies. In real-world games a strategy is often dominated by another strategy. A dominated strategy gives worse or at most the same utility as the strategy dominating it. Therefore it never makes sense to play a dominated strategy. The search space is made smaller by using iterated removal of dominated strategies before looking for the Nash equilibrium in the support.

3 Problem Setting

A typical action game can be described as a set of combat encounters between the PC and multiple enemy NPCs. The NPCs usually try to move close to the PC and then make their attack. The abstract test simulation models one such encounter omitting the final attack phase. The purpose of the simulation is therefore to observe NPC movement in a game-like situation, in which the NPCs try to get close to the PC while avoiding being exposed to the PC's possible weapons. A starting point for the simulation is described in Figure 1. The game area consists of squares. In the figure the PC is represented by 'P' and the NPCs by numbers from 1 to 3. 'X' represents a wall square and '.' represents open ground.

```
.............................
.............................
.............XX..............
......1.....X.....P..........
.........XXXX................
...2..XXXXX..................
.......XX....................
.............................
.............................
....X........................
....XXX..............XXXXX...
...........X..........XXXX...
.......................X.....
......................3......
```

Figure 1: A starting point for test simulation.

The test simulation has a set of rules. It is divided into turns, and the NPCs decide their actions each turn. The PC doesn't move in this simulation. Each NPC has a set of five possible actions, which can be performed during a single turn: { left, right, up, down, do nothing }. A NPC cannot move through the walls, so whenever a NPC is located next to a wall, it's action set is reduced by the action that would move it into the wall. The PC cannot see through walls. Two NPCs can be located in the same square, but a NPC cannot move to the square occupied by the PC.

In the test simulation all NPCs choose their actions simultaneously from their action sets. However, the Nash equilibrium search is not made by the NPCs themselves, but by a central agency. This is justified, since the chosen strategies form a Nash equilibrium. Even if the computing was left to the NPCs, their individual strategy choices would still converge to a Nash equilibrium.

Levy and Rosenschein (1992) studied the game theoretic solution for the predator-prey problem domain, which has four predator agents trying to encircle a prey agent in a grid. They used game theory as a means to have the predator agents move more precisely when they got near the prey. The game theoretic predators avoided collisions by selecting game states where no collisions occured: these were naturally the Nash equilibria. In our simulation, however, the focus is different. Game theory allows the NPCs to act individually rationally in a way that can be, if necessary, against other agents and their goals. This allows for realistic-looking cooperation, which in turn leads to a greater immersion for the player. It is easy to make a NPC in a computer game to be more dangerous for the PC: the NPC can be made, for instance, faster or stronger. The problem of making the NPCs appear life-like is much more difficult. The simulation aims to provide some insight into how realistic-looking NPC movement might be attained.

All "intelligence" of the NPCs is encoded in the utility function, which means that all targeted behavioral patterns must be present within it. Levy and Rosenschein used a two-part utility function with the predator agents: one part of the function gives the agents payoff for minimizing the distance to the prey agent and the other half encourages circulation by rewarding blocked prey movement directions. Following the same principle, the utility function that we use is made of terms that detail different aspects of the NPCs' goals. Each term has an additional weight multiplier, which is used to control and balance the amount of importance the term has. We found the following terms to be the most important in guiding the NPCs' actions in the simulation: aggression, safety, balance, ambition, personal space and inertia. The final utility function is the sum of all the terms weighted with their weight multipliers. The terms are detailed below.

Aggression means the NPC's wish to get close to the PC. The term's value is $-xD_i$, where D_i is the NPC i's distance from the player and x is an additional constant multiplier representing the growing of aggression when the NPC gets nearer the PC.

Safety represents the NPC's reluctance to take risks. When a NPC is threatened, it gets a fine. The fine amounts to $-r_i/k$, where r_i is the severity of the threat that the NPC encounters and k is the number of NPCs who are suspect to the same threat. If k is zero, no NPCs are threatened, and the term is not used. In the test simulation a NPC was threatened, if it was on the PC's line of sight. The divisor is used to make it safer for one NPC, if several NPCs are subjected to the same threat from the PC.

Balance is the NPC's intention to stay approximately at the same distance from the PC as the other NPCs. The value of balance is $-d_i$, where $d_i = \left|D_i - \frac{\sum D_{-i}}{k}\right|$. $\frac{\sum D_{-i}}{k}$ is the average distance of all other NPCs from the PC. D_i is the NPC i's distance from the PC. $\left|D_i - \frac{\sum D_{-i}}{k}\right|$ gets bigger as the NPC's distance from the PC gets further from the average distance. The term is needed to make the NPCs try to maintain a steady and simultaneous advance towards the PC, and to prevent the NPCs from running to the PC one by one. The term is not used if there are no other NPCs in the game.

Ambition means the NPC's desire to be the first attacker towards the PC. If the NPC is not moving towards the PC, ambition value is 0. If no NPC is going towards the PC, the ambition value is tx, where t it the amount of turns in which no NPC has moved towards the PC and x is constant.

Personal space is the amount of personal space a NPC needs. The term has value x_i, which is the NPC's distance to the nearest other NPC, if the distance is below a threshold value x'. If $x_i > x'$, the term has value x' instead of x_i. This term is needed to avoid the NPCs packing together and encourage them to go around obstacles from different sides.

Inertia is the NPC's tendency to keep to a previously selected action. The term makes the NPCs appear consistent in their actions. If the NPC is moving in the same direction as in the previous game turn, it gets bonus x. If the NPC is moving in an orthogonal direction, it gets the bonus $\frac{x}{2}$. Inertia helps to prevent the NPCs from reverting their decisions: if ambition drives the NPCs from their hiding places, inertia keeps them from retreating instantly back into safety.

Since the utility function provides the information about which actions the NPC values, all other necessary AI functions must be implemented there. The test simulation required implementation of A* algorithm for obstacle avoidance: all measured distances towards the PC or other NPCs are actually shortest-path distances. Game-specific values can also be adjusted in the utility function. For instance, the game designer may want to change the game difficulty level mid-game by tweaking the game difficulty parameters (Spronck et al., 2004). These adjustments must be made within the utility function, otherwise they have no effect in NPC action selection.

The simple search algorithm is deterministic by nature, and therefore the Nash equilibrium found from any given starting setup is always the same. If a mixed strategy is found, randomness follows implicitly, because the action is randomly selected from the probability distribution. However, the algorithm is biased towards small supports for efficiency reasons, and therefore tends to find pure strategies first. The pure strategies are common with the game simulation setting, since the NPCs are rarely competing directly against one another. The game designer may want to implement randomness in the utility function by adding a new term, *error*, which is a random value from $[0 \ldots 1]$. The weight multiplier can be used to adjust the error range.

4 Results

The simulation yielded two kinds of results: the time needed to find the Nash equilibrium during a typical game turn and the NPCs' actions using the utility function described in section 3. The simulation run began from the setting in Figure 1.

Because the computer game AI is only good if it gives the player a sense of immersion, the evaluation of the utility function's suitability must be done from the viewpoint of game playability. However, no large gameplay tests were organized. The utility function goodness is approximated by visually inspecting the game setting after every game turn.

Figure 2 shows the game board on turn 3. The '+'-signs represent the squares that the NPCs have been in. NPC 2 began the game by moving south, even though its distance to the PC is the same in both the northern and southern route around the obstacle. This is due to the fact that NPCs 1 and 2 were pushed away from each another by the term *personal space* in the utility function.

Figure 3 on turn 10 has all NPCs in place to begin the final stage of the assault. None of the NPCs are on the PC's line of sight. The NPCs 3 and 2 have reached the edge of the open field before NPC 1, but they have elected to wait until everyone is in position for the attack. The term *safety* is holding them from attacking over the open ground.

```
.............................
.............................
.............XX..............
......+++1..X.....P..........
.........XXXX................
...+..XXXXX..................
...+...XX....................
...+2........................
.............................
....X........................
....XXX..............XXXXX...
...........X.........3XXXX...
.....................++X.....
......................+......
```

Figure 2: Turn 3. NPC 1 has begun to go around the obstacle by North and NPC 2 from South. NPC 3 has moved into cover.

```
.............................
...........+++1..............
...........+.XX..............
......++++++X.....P..........
.........XXXX................
...+..XXXXX..................
...+...XX....................
...+++++2....................
.............................
....X........................
....XXX..............XXXXX...
...........X.........3XXXX...
.....................++X.....
......................+......
```

Figure 3: Turn 10. All members of the NPC team have arrived to the edge of the open ground.

Game turn 12 is represented in Figure 4. The NPCs have waited one turn in place and then decided to attack. Each NPC has moved simultaneously away from cover towards the PC. In game theoretic sense, two things might have happened. The first possibility is that term *ambience* makes one NPC's utility from moving towards the PC grow so big that attacking dominates the NPC's other possible actions. Therefore the NPC's best course of action is to attack regardless of the other NPCs' actions. When this happens, the sanction from term *safety* diminishes due to the number of visible NPCs, and it suddenly makes sense for the other NPCs to participate in the attack. The other possibility is that the algorithm for finding Nash equilibria has found the equilibrium, in which all NPCs move forward, before the equilibrium, in which the NPCs stay put.

NPCs succeed in synchronizing their attack be-

```
.............................
...........++++1.............
...........+.XX..............
......++++++X.....P..........
.........XXXX................
...+..XXXXX..................
...+...XX....................
...++++++2...................
.............................
....X........................
....XXX..............XXXXX...
...........X........3+XXXX...
.....................++X.....
......................+......
```

Figure 4: Turn 12. NPC team decides to attack. All NPCs move simultaneously away from cover.

```
.............................
...........++++++++..........
...........+.XX...1..........
......++++++X.....P..........
.........XXXX.....3..........
...+..XXXXX.......+..........
...+...XX2........+..........
...+++++++........+..........
..................+..........
....X.............+..........
....XXX...........+..XXXXX...
...........X......++++XXXX...
.....................++X.....
......................+......
```

Figure 5: Turn 21. NPCs 1 and 3 have reached the PC. NPC 2 has gone into hiding once more.

cause the Nash equilibrium defines the strategies for all NPCs before the real movement. Synchronization leaves the human player with the impression of planning and communicating enemies. If the NPCs had ran into the open one by one, stopping the enemies would have been much easier for the human player, and the attack of the last NPC might have seemed foolhardy after the demise of its companions.

Figure 5 shows the game board on turn 21. NPCs 1 and 3 have moved next to the PC. NPC 2 has found new cover and has decided not to move forward again. The situation seems erroneous, and it is true that the NPC 2's actions seem to undermine the efficiency of the attack. However, this can be interpreted to show that NPC 2 has reexamined the situation and decided to stay behind for it's own safety. One way to resolve the situation is to change the term *safety* to lessen the threat value, if at least one NPC is already in hand-to-hand combat with the PC.

Creating a usable utility function is quite straightforward if agent's goals can be determined. Balancing the terms to produce the desired behavior patterns in different game settings can be more time-consuming. Each different video game needs a specific utility function for its NPCs, since for instance the distance measurements are done in different units. Also the game designer may want to introduce different behavior to different NPC types, which is done by creating a utility function for each NPC class within the video game.

The test simulation used the McKelvey et al. (2006) implementation of the previously mentioned simple search algorithm. The algorithm's time complexity limits the simulation's feasibility when the number of agents in the game grows larger.

We measured the time needed for finding a single Nash equilibrium in a Macintosh computer equipped with a 2.1 GHz processor. The extra calculations in the utility function (such as A* algorithm) were left out. The needed time was calculated using the three-agent setting described in Figure 1 and the previously detailed utility function. The measurements were also made using six NPCs, whose starting positions are detailed in Figure 6.

```
.............................
.............................
.............XX..P...........
......1.....X................
.........XXXX................
...2..XXXXX...............5..
.......XX....................
.................4...........
.............................
....X........................
....XXX..............XXXXX...
...........X..........XXXX...
....6..................X.....
......................3......
```

Figure 6: The starting point for the six-player simulation.

Both games were run for 25 turns, and the experiment was repeated ten times. The games had therefore 250 data points each. The test results are presented in table 1.

Agents	min.	max.	avg.	median
3	42 ms	225 ms	71 ms	62 ms
6	97 ms	2760 ms	254 ms	146 ms

Table 1: Experimental results for the time needed to find the first Nash equilibrium in attack game.

The results show that in a three-player game the Nash equilibrium was found in average within 71 milliseconds. This can be still feasible for real video games. In six-player games the worst calculation took almost three seconds, which is far too long for action games. Still, the median in six-player game was only 146 milliseconds, which may still be acceptable. In both games a mixed strategy was never needed: the Nash equilibria were always found in supports of size 1. The worst times were measured in the first turn. When a NPC had several dominated actions or was next to a wall, the search was faster, because the search space was reduced.

5 Conclusion

Using game theoretic methods in action games seems a promising way to do a more plausible NPC group AI. If the agents try to find a Nash equilibrium, their individual decisions are rational. If the agents' utility functions are designed to find cooperative behavior patterns, the group seems to function with a degree of cooperation. Having agents do their decisions based on agents' internal valuations helps agents maintain their intelligent-looking behavior in situations, where a scripted approach would lead to non-satisfactory results. This might make the game designer's work easier, since all encounters between the PC and a NPC group need not be planned in advance.

The problem in this approach is the time complexity of finding Nash equilibria. Finding one equilibrium is difficult, and finding all equilibria is prohibitively expensive. Still, our results indicate that modern computers with a good heuristic might be able to find one Nash equilibrium relatively fast, especially if the NPCs' action sets are limited and the number of NPCs is small. If the Nash Equilibria cannot be found soon enough, it is up to the game designer to decide the fallback mechanism. The scripted approach is one possibility, and another is to use a greedy algorithm based on the utility functions. A greedy algorithm would not waste time on calculating the possible future actions of the other NPCs, but would assume that the NPCs stayed idle or continued with a similar action as in the previous game round.

References

Vincent Conitzer and Tuomas Sandholm. Complexity results about Nash equilibria. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, pages 765–771, 2003.

Prajit K. Dutta. *Strategies and Games: Theory and Practice*. MIT Press, 1999. ISBN 0262041693.

Ran Levy and Jeffrey S. Rosenschein. A game theoretic approach to the pursuit problem. In *Proceedings of the Eleventh International Workshop on Distributed Artificial Intelligence*, pages 195–213, Glen Arbor, Michigan, February 1992.

Richard D. McKelvey and Andrew M. McLennan. Computation of equilibria in finite games. In H. Amman, D. Kendrick, and J. Rust, editors, *Handbook of Computational Economics*, volume 1, pages 87–142. Elsevier, 1996.

Richard D. McKelvey, Andrew M. McLennan, and Theodore L. Turocy. Gambit: Software tools for game theory, version 0.2006.01.20, 2006. URL http://econweb.tamu.edu/gambit/.

Alexander Nareyek. Review: Intelligent agents for computer games. In T. Anthony Marsland and Ian Frank, editors, *Computers and Games*, volume 2063 of *Lecture Notes in Computer Science*, pages 414–422. Springer, 2000. ISBN 3-540-43080-6.

John Nash. Equilibrium points in n-player games. In *Proceedings Of NAS 36*, 1950.

Christos H. Papadimitriou and Tim Roughgarden. Computing equilibria in multi-player games. In *SODA*, pages 82–91. SIAM, 2005. ISBN 0-89871-585-7.

Ryan Porter, Eugene Nudelman, and Yoav Shoham. Simple search methods for finding a Nash equilibrium. In Deborah L. McGuinness and George Ferguson, editors, *AAAI*, pages 664–669. AAAI Press / The MIT Press, 2004. ISBN 0-262-51183-5.

Steve Rabin. Common game AI techniques. In Steve Rabin, editor, *AI Game Programming Wisdom*, volume 2. Charles River Media, 2003.

Pieter Spronck, Ida G. Sprinkhuizen-Kuyper, and Eric O. Postma. On-line adaptation of game opponent AI with dynamic scripting. *Int. J. Intell. Games & Simulation*, 3(1):45–53, 2004.

Milind Tambe. Towards flexible teamwork. *J. Artif. Intell. Res. (JAIR)*, 7:83–124, 1997.

Higher Order Statistics in Play-out Analysis

Tapani Raiko[⋆]

[⋆]Adaptive Informatics Research Centre
Helsinki University of Technology
tapani.raiko@tkk.fi

Abstract

Playing out the game from the current state to the end many times randomly, provides statistics that can be used for selecting the best move. This play-out analysis has proved to work well in games such as Backgammon, Bridge, and (miniature) Go. This paper introduces a method that selects relevant patterns of moves to collect higher order statistics. Play-out analysis avoids the horizon effect of regular game-tree search. The proposed method is especially effective when the game can be decomposed into a number of subgames. Preliminary experiments on the board games of Hex and Y are reported.

1 Introduction

Historically, chess has been thought to be a good testbed for artificial intelligence. In the last 10 years, though, computers have risen above human level. This is largely due to fast computers being able to evaluate a huge number of possibilities. One can argue, whether these programs show much intelligence after all.

There are still games that are similar to chess in some sence but have proved to be difficult for computer players. These games include Go, Hex, Y, and Havannah. The games have some common factors: First, the number of available moves at a time is large, which prevents the use of brute-force search to explore all the possibilities as the search tree grows exponentially with respect to depth. Second, there are lots of forcing moves which boost the so called horizon effect. If the program sees a limited number of moves ahead (search depth), there is often a way to postpone the interesting events beyound that horizon.

This paper introduces a method that avoids these two pitfalls. The search is performed by playing out games from the current situation to the end several times randomly. This is known as play-out analysis or Monte Carlo playing. The number of explored possibilities is constant with respect to search depth. Also, as the games are played out all the way to the end, there is no need to evaluate a game in progress, thus avoiding the horizon effect.

1.1 Game of Hex

Hex is a two-player boardgame with no chance. It is played on an $n \times n$ rhombus of hexagons. The players have colours red and blue and the edges of the board are coloured red and blue such that parallel edges have the same colour. The game is started with an empty board and the players alternately fill a cell with a piece of ones own colour. Red player wins if the red edges are connected with a path of red cells and vice versa. When the board fills up, exactly one of the players has formed a connecting path, see Figure 5 for an example game.

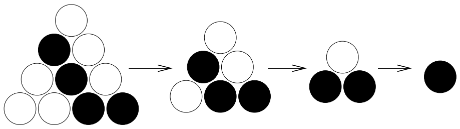


Figure 1: A small filled Y board is shown to be won by black by micro reductions.

1.2 Game of Y

The game of Y is a two-player boardgame with no chance and a relative to the more popular Hex. Y is played on a triangular board formed of hexagonal cells that are empty in the beginning. Two players alternately fill empty cells with pieces of their own colour. The player who creates an unbroken chain of pieces that connect all three edges, wins. Note that corners belong to both edges. The standard board in Figure 2 is slightly bent with three of the hexagons replaced by pentagons.

The fact that Y cannot end in a draw on a straight board can be proven using so called micro reductions described in (van Rijswijck, 2002). Size n board is reduced to a size $n-1$ board where each cell on the

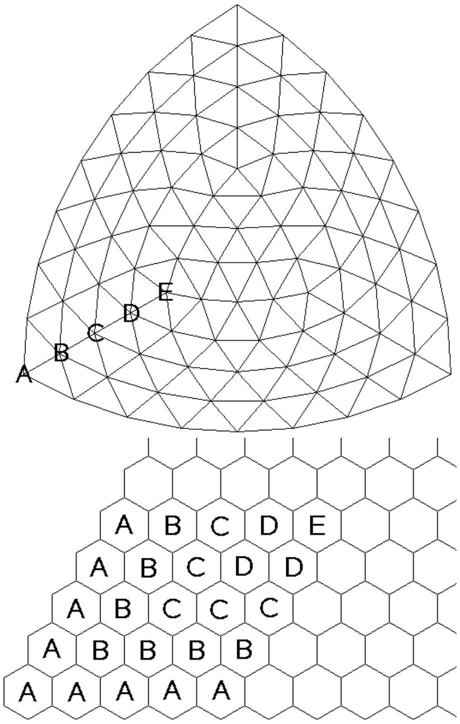


Figure 2: Top: The game of Y is usually played on the bent board, on points rather than cells. Bottom: The bent board can be transformed into a straight one for analysis such as micro reduction. Filling a point in the bent board fills all the cells with the same letter in the straight board. Only one corner is shown.

reduced board gets the majority colour of the three nearest cells on the original board. An example is given in Figure 1. It turns out that if a chain touches a side of the board, it does so also on the reduced board. A winning chain will thus be a winning chain on all the smaller boards including the trivial board of size 1. Figure 2 shows how the same analysis can be done for the bent board by first transforming it into a straight one.

1.3 Related Work

Abramson (1990) proposed the expected-outcome heuristic for game state evaluation. The value of a game state is the expected outcome of the game given random play from that on. The value can be used to evaluate leaf nodes in a game-tree that has the current state as the root node and possible moves as children of the node. The expected-outcome heuristic applies to a large number of games with or without randomness or hidden information and with one or more players. It is best applicable to games that always end after a reasonable number of moves, such as the games of Hex and Y. Expected outcome heuristic is also known as roll-out analysis or play-out analysis.

In many planning problems and games, the order of moves or actions is not always important. This applies especially in games of Hex and Y where the order does not have any role as long as the same moves are made. In Monte Carlo playing, unimportance of the order means that the all-moves-as-first heuristic (Brügmann, 1993) works well: After a game has been played out, not only the evaluation of the first move is changed, but all the subsequent moves are handled as they were the first move. Thus, in the games of Hex and Y, instead of playing the game out, one can just fill the empty cells by pieces of random colour. Sampling colours for the empty cells then corresponds to Monte Carlo playing. Those cells that were filled by the winning player, are important.

All-moves-as-first heuristic is very good in reducing the number of random samples compared to basic expected outcome, but unfortunately it limits the lookahead into just one move in some sense. Bouzy (2004) proposed a hybrid between expected-outcome and all-moves-as-first heuristics. A shallow game tree is made where each leaf is evaluated using the all-moves-as-first heuristic.

Müller (1999) proposed decomposition search with an application to Go endgames. The idea is that the game can be decomposed into a sum of subgames which are somewhat independent of each other. Because the subgames are small, the full game tree can be constructed for each of them, including local pass moves. Then, many of the moves can be pruned for instance because they are dominated by other moves in the same subgame. The global search is thus sped up a lot, allowing perfect solution of end-games of dozens of moves. A similar approach is suggested to be used heuristically for middle-game where the division into subgames is not yet strict.

Already Brügmann (1993) suggested an extension of the all-moves-as-first heuristic into an higher order heuristic, where not just the value of each move is evaluated but the value of making a move after some other N moves have been made. Time was not ripe for higher order considerations for computational resources. If there are M moves available, one would have to collect enough statistics for M^N combinations of moves. Now, thirteen years later, it would still be a waste of resources to try to uniformly estimate all combinations.

Inductive logic programming (ILP) (Muggleton and De Raedt, 1994) involves finding patterns of increasing size in data. A pattern in this case is a set of moves (in addition to the ones that have been already made) and the data are the random games. One can estimate the value of a move given that the pattern has already been played. New patterns are created by refinement operators, in this case by adding a move to an existing pattern. By using ILP, we can make the higher-order heuristic selective, that is, we do not have to estimate all M^N combinations of moves but only MP where P is the number of patterns.

2 Higher order statistics in play-out analysis

Black and white player are playing a game with alternate moves. At some state s of the game, a play-out analysis means that k hypothetical continuations c of the game are played out to the end and statistics from these play outs are used to determine the next move. The play outs might be sampled by drawing moves from a uniform distribution over all legal moves, or by other means. To sample a semi-random move, one can add noise from a random number generator to a heuristic evaluation of each move and select the best one.

The expected-outcome heuristic h_e evaluates the move m_0 as

$$h_e(m_0) = \frac{\sum_{i=1}^{k} \chi(c_i \text{ is a win})\chi(m_0 \text{ first move of } c_i)}{\sum_{i=1}^{k} \chi(m_0 \text{ first move of } c_i)}, \tag{1}$$

where χ is the indicator function ($\chi(\text{true}) = 1$ and $\chi(\text{false}) = 0$). The best move is thus the one that lead to most wins in the play outs c.

Note that in Hex and Y the order of moves does not matter. One can play the game from current state all the way to filling the board. The winner of the game can be determined from the filled board without knowing the order of moves. One can think that the single play out represents all the possible play outs that lead to the same position. This leads to the all-moves-as-first heuristic h_a:

$$h_a(m_0) = \frac{\sum_{i=1}^{k} \chi(c_i \text{ is a win})\chi(m_0 \text{ is made in } c_i)}{\sum_{i=1}^{k} \chi(m_0 \text{ is made in } c_i)}. \tag{2}$$

The best move is the one that lead to wins when used at any point in the continuation. An example is shown in Figure 3. Often, play out is done using a semi-random move selection with the heuristic itself.

Higher order heuristic h_h evaluates the move m_0 after moves $m_1, m_2, \ldots, m_l$ are made in any order.

$$h_h(m_0 \mid \{m_j\}_{j=1}^{l}) = \tag{3}$$

$$\frac{\sum_{i=1}^{k} \chi(c_i \text{ is a win})\chi(m_0 \text{ after } \{m_j\}_{j=1}^{l} \text{ in } c_i)}{\sum_{i=1}^{k} \chi(m_0 \text{ after } \{m_j\}_{j=1}^{l} \text{ in } c_i)}.$$

The sets of moves $\{m_j\}_{j=1}^{l}$ are called patterns. Many different patterns apply in a future state when sampling a play out, and one must decide how to combine evaluations corresponding to different patterns. Also, one has to be selective which patterns are taken into consideration. Before going into these details, an example is shown in Figure 4.

2.1 Combining the evaluations by different patterns

Many patterns can apply to the same state, that is, the moves of more than one pattern have been made. It would be possible to construct complicated rules for combining their evaluations; e.g. the average of the evaluations of applying patterns weighted with a function of the pattern size. One should note that play out is the most time intensive part of the algorithm and combining happens at every play-out move so it has to be fast. The proposed combining rule is:

- The maximum of evaluations given by each pattern that applies but is not a subpattern of another applying pattern.

The motivation is that the more specific pattern gives more accurate information about a move than its subpattern. In Figure 4, patterns 1, 2, and 4 apply to the example state, but pattern 1 is a subpattern of the other two so it is not taken into account. Using the maximum helps in computational complexity. One can take the maximum first within a pattern and then over different patterns. The evaluations of a pattern can be stored in a heap to make the time complexity logarithmic with respect to the number of available moves.

2.2 Selection of patterns

An algorithm based on inductive logic programming is used to select patterns. The process starts with just the empty pattern (for which $l = 0$). Repeatedly, after a certain number of play outs, new patterns are added. An existing pattern generates candidates where each possible m_0 is added to the pattern, one at a time. The candidate is accepted if it has appeared often enough, and if m_0 is a relevant addition to the pattern. If the original pattern makes move m_0 more valuable than without the pattern, the moves seem to be related. The used criterion was based the mutual information between the made move and winning, given the pattern. If the move makes winning less likely, the criterion was set to zero. Then the maximum of mutual informations of the move and winning, given any of the subpatterns, is substracted from the score. When new patterns are added, statistics are copied from its parent with a small weight. Also, the statistics of the first play-outs are slowly forgotten by exponential decay.

2.3 Exploration and exploitation

One has to balance between exploring lots of truly different play outs and exploiting the known good moves and studying them more closely. Here it is done by using simulated annealing: In the beginning, the amount of noise is large, but towards the actual selection of the move, the amount of noise is decreased to zero linearly.

Another possibility would be to add a constant (say 1) to the numerator and the denominator of Equation 3 for emphasising exploration of unseen moves. This corresponds to an optimistic prior or pseudocount. When the time comes to select the actual move, one wants to de-emphasise unseen moves, which is done by adding the constant only to the denominator.

2.4 Final move selection

To select the best move after play-out analysis is done, it is possible to simply pick the move m_0 with the highest heuristic value:

$$\arg\max_{m_0} h_h(m_0 \mid \{\}). \tag{4}$$

This is the only option with first-order heuristics, but higher order heuristics allow for more. One can make a min-max tree search in the tree formed by known patterns. For instance, using the second-order heuristics, one selects the move m_1, for which the best answer m_0 by the opponent is the worst:

$$\arg\min_{m_1}\max_{m_0} h_h(m_0 \mid \{m_1\}). \tag{5}$$

2.5 Summary of the algorithm

Given a state $s(0)$ select a move $m(0)$ by

```
1: Patterns $P = \{\{\}\}$, # of play out $k = 0$
2: Play-out depth $j = 0$
3: Move $m(j) = \arg\max_m h_h(m \mid p) + \text{noise}$
4: Make move $m(j)$ in state $s(j)$ to get new $s(j+1)$
5: Increase $j$ by one
6: If $s(j)$ not finished, loop to 3
7: Increase $k$ by one
8: Save moves $m(\cdot)$ and the result of $s(j)$ in $c_k$
9: Loop to 2 for some time
10: Add new patterns to $P$
11: Loop to 2 for some time
12: $m(0) = \arg\min_{m_1} \max_{m_0} h_h(m_0 \mid \{m_1\})$
```

On line 3, the moves of the pattern $p \in P$ and $p \subset \{m(0), \ldots, m(i-1)\}$ and there must be no $q \in P$ such that $p \subset q \subset \{m(0), \ldots, m(i-1)\}$. On line 10, the candidate patterns are all patterns in P extended by one move, and they are accepted if the particular move has been selected by that pattern more than some threshold number of times on line 3.

3 Experiments

The proposed method was applied to the games of Hex and Y. An implementation is available at http://www.cis.hut.fi/praiko/hex/. Figure 5 shows a game of Hex played by the proposed method against itself. The red player won as can be seen from the path starting from moves 43, 33, 21. The level of play is not very high, but taking into consideration that the system is not game-specific and that the implementation is preliminary, the level is acceptable.

Figure 6 shows the position before move 31. Because of symmetry of the criterion for selecting new patterns, they are generated in pairs, for instance, the patterns red at a and blue at a were the first two. Patterns in the order of appearance $\{\}$, $\{a\}$, $\{c\}$, $\{d\}$, $\{h\}$, {blue c, f}, {red a, b}, {red c, d}, {red d, c}, {blue a, g}, and so forth. The patterns are clearly concentrated on areas of interest and they are local.

A tournament between 8 different computer players was held in the game of Y such that each player met every opponent 10 times as black and 10 times as white, 640 games in total. The first player A made just random moves chosen uniformly from all the available moves. Three players B, C, and D used the all-moves-as-first heuristic with 100, 1000, and 10000 fully random play-outs per move, accordingly. Players E and F used second-order heuristics, that is, considering all patterns where a single move is made by the player in turn. The final two players G and H used selective patterns, whose number varied from 0 to 99 and size from 0 to 7. The players E–H used

player	order	# playouts	time/ms	win %
A	-	0	5	0
B	1	100	8	16
C	1	1000	36	66
D	1	10000	276	65
E	2	1000	125	65
F	2	10000	1155	61
G	N	1000	240	64
H	N	10000	14735	63

Table 1: The number of playouts, the order of used statistics, the average thinking time in milliseconds per move, and the winning percentage against other players is given for each player.

	A	B	C	D	E	F	G	H
A	7	0	0	0	0	0	0	0
B	10	6	0	1	1	0	0	0
C	10	10	7	3	7	6	6	6
D	10	10	5	2	5	5	5	8
E	10	10	7	7	7	6	3	6
F	10	10	6	6	4	4	6	3
G	10	10	4	5	5	6	8	6
H	10	9	4	5	6	7	6	6

Table 2: Wins out of ten for the black player (label on the left) against the white player (label on the top).

the second-order move selection in Equation (5). The number of play-outs and the average time per move is shown in table 1.

The results are shown in Table 2. Black won 53% of the matches since the first move gives an advantage. Player A lost all games againt other players. Player B was also clearly worse than the others. The differences between the players C–H are not clear from the limited amount of data.

4 Discussion

The experiments did not show improvement over first-order heuristics. This might be due to a larger need for samples, or perhaps the criteria for selecting new patterns were unoptimal. Baum and Smith (1997) propose a well-founded measure of relevance of expanding a leaf in a search tree when evaluations are probabilistic. This measure could be applied here as well. There is a lot of room for other improvements in general and in reducing computational complexity as well as taking into use application-specific heuristics.

The applicability of the proposed approach is limited to games where the order of moves is not very important as long as all of them are made. In Hex and Y this applies exactly, whereas in games of Havannah and Go, it applies approximatively. Perhaps patterns with more complicated structure could be introduced, such as any boolean formula over moves and order relations among moves.

The random filling heuristic in the game of Hex corresponds to communication reliability in graph theory. Two-terminal network reliablity (Brecht and Colbourn, 1988), or probabilistic connectedness, is the probability that two nodes in a network can communicate. Edges are assumed to have statistically independent probabilities of failing but nodes are assumed reliable. The studies in graph theory have brought results such as that the exact calculation of two-terminal reliability requires exponential computations.

The random filling heuristic for the game of Y is used by van Rijswijck (2002). He uses implicit random colouring of empty cells but instead of sampling from the distribution, he uses micro reduction repeatedly until the single value on the size-1 board can be directly used as a heuristic. At each step of the reduction, all the probabilities are assumed to be independent, which makes the heuristic quick and dirty. The transformation in Figure 2 allows the usage of this heuristic also on the bent board. It would be interesting to test this against the proposed approach.

The proposed method generalises trivially to all 1-or-more-player games and to other rewards than win/loss. The 1 player version has a connection to nonlinear planning (see book by McAllester and Rosenblitt, 1991). Planning aims at finding a sequence of actions that lead to some goal. Sometimes plan involves non-interacting steps that can be reordered. This is taken into account in a nonlinear plan which only contains a partial order among actions. The set of patterns and statistics can be interpreted as a nonlinear plan.

The idea of estimating statistics for growing patterns is also used in natural language processing by (Siivola and Pellom, 2005). The model probabilistically predicts the next word given a context or pattern of n previous words. New patterns are generated by adding a word to an existing pattern. For the final model, some of the patterns are pruned.

Currently all the patterns and statistics are forgotten after each move, while they could still be used. Also, the proposed system could be used with machine learning. Patterns reappear from game to game so life-long learning should be possible. Games such as Hex, Y, Go, and Havannah all have also limited translational invariance which could be used for generalisation. Machine learning in this setting is left as future work.

5 Conclusion

One can collect statistics from playing out the game randomly to the end many times. This paper proposed a method for a selective collection of higher order statistics, that is, evaluations of some combinations of moves. If the game can be divided into a number of subgames, the proposed system seems to be able to find relevant combinations concentrated on a single subgame at a time. The preliminary experiments did not yet show significant improvement over the first-order approach, but a door has been opened for further improvement.

The paper also gave some analysis on the game of Y. The proof of impossiblity of draws was extended to cover the bent board. A quantitative difference between the straight and the bent board with respect to the importance of the centre was shown. Also, playout analysis was applied to the games of Hex and Y for the first time.

Acknowledgements

The author wishes to thank Jaakko Peltonen and Teemu Hirsimäki for useful discussions. This work was supported in part by the Finnish Centre of Excellence Programme (2000-2005) under the project New Information Processing Principles and by the IST Programme of the European Community under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

References

Bruce Abramson. Expected-outcome: A general model of static evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(2): 182–193, February 1990.

Eric B. Baum and Warren D. Smith. A bayesian approach to relevance in game playing. *Artificial Intelligence*, 97(1–2):195–242, 1997.

Bruno Bouzy. Associating shallow and selective global tree search with monte carlo for 9x9 go. In *Proceedings of the 4th Computer and Games Conference (CG04)*, pages 67–80, Ramat-Gan, Israel, 2004.

Timothy B. Brecht and Charles J. Colbourn. Lower bounds on two-terminal network reliability. *DAMATH: Discrete Applied Mathematics and Combinatorial Operations Research and Computer Science*, 21:185–198, 1988.

Bernd Brügmann. Monte Carlo Go, 1993. Available at `http://www.cgl.ucsf.edu/go/Programs/`.

David McAllester and David Rosenblitt. Systematic nonlinear planning. In *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91)*, volume 2, pages 634–639, Anaheim, California, USA, 1991. AAAI Press/MIT Press. ISBN 0-262-51059-6.

Stephen Muggleton and Luc De Raedt. Inductive logic programming: Theory and methods. *Journal of Logic Programming*, 19/20:629–679, 1994.

Martin Müller. Decomposition search: A combinatorial games approach to game tree search, with applications to solving Go endgames. In *Proceedings of the International Joined Conference on Articial Intelligence (IJCAI 1999)*, volume 1, pages 578–583, 1999.

Vesa Siivola and Bryan Pellom. Growing an n-gram language model. In *Proceedings of the 9th European Conference on Speech Communication and Technology (Interspeech'2005)*, Lisbon, Portugal, 2005.

Jack van Rijswijck. Search and evaluation in Hex. Technical report, Department of Computing Science, University of Alberta, 2002.

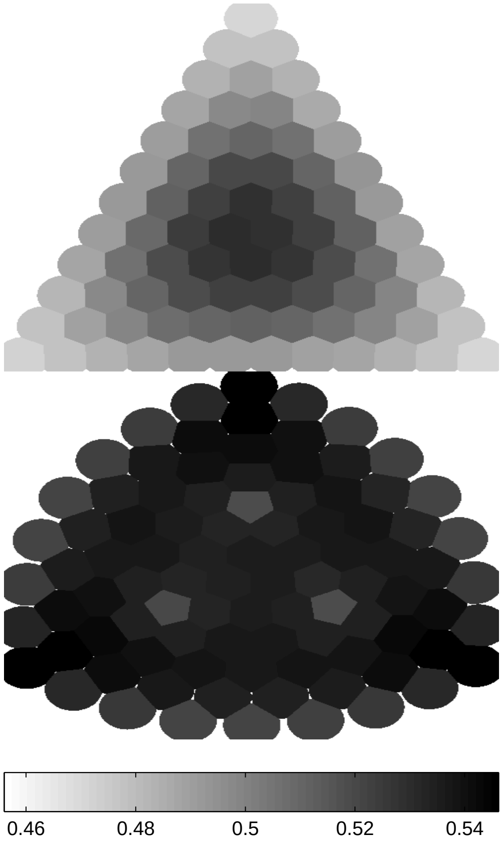


Figure 3: The expected-outcome, or equivalently in this case, the all-moves-as-first heuristic for the first move on a straight (top) and a bent (bottom) Y board. The colour shows the probability of a black win assuming random play after the first move. Note that the straight board gives a large emphasis on the center, which is the reason why the bent board is often used instead.

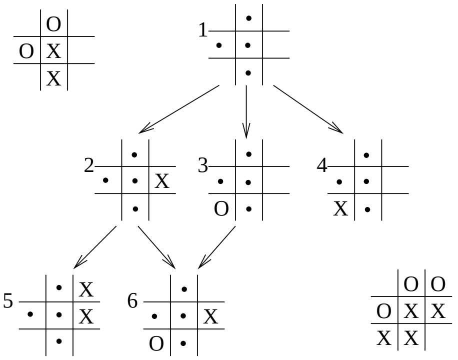


Figure 4: The current game state in tic-tac-toe is shown at the top left corner. Six patterns are numbered and shown in a graph. When playing out, a state at the bottom right corner is encountered. Patterns 1, 2, and 4 apply to it.

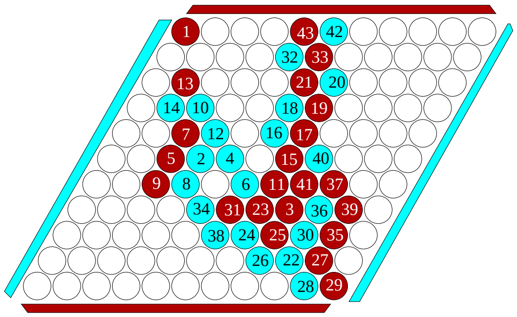


Figure 5: An example game of Hex by self-play of the proposed system.

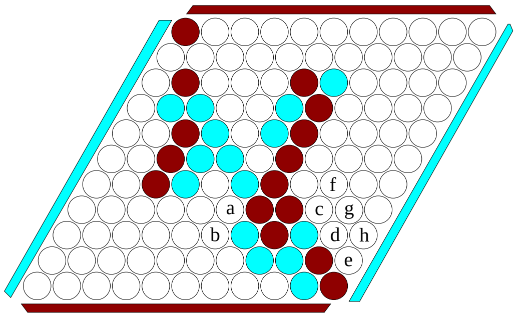


Figure 6: At move 31, the first patterns use the cells marked with $a, b, \ldots, h$. See text for explatations.

Evolving poker players using NeuroEvolution of Augmenting Topologies

Morten Lynge Jensen[*†]
[*]Department of Computer Science
University of Aarhus, Denmark
U032933@daimi.au.dk

Abstract

Evolving poker players is not a new idea. Many attempts have been made, using different approaches. This article describes an experiment attempting to improve the evolution of neural networks capable of playing the poker variant Texas Hold'em at tables with 5 opponents by a divide and conquer approach combined with introduction of subpopulations to distribute the algorithm to multiple machines. This method tries to challenge the individuals in different ways in order to identify properties of interest, and the results show potential for this method to develop strong players.

1 Introduction

When using fixed topologies for neural networks, one face the problem of sacrificing obtained knowledge when improving the strategy, since altering the network may change knowledge obtained in previous generations. Thereby the fixed topology favours the most significant parameters for the strategy, leading to the loss of less significant information. This less significant information might be useful in some decisions, and should therefore be kept within the topology.

The NEAT method combats this problem by making the topology more complex if necessary. Stanley and Miikkulainen (2002) show that applying the NEAT method for evolving neural networks to a competitive simulated robot duel domain leads to more complex networks as evolution progresses, where the complexification elaborates on existing strategies and the evolved strategies are more sophisticated compared to networks with fixed topologies.

Poker is a probabilistic game with imperfect information, where players can achieve good results by exploiting their opponents' weaknesses. The rules are simple, but the game contains aspects of opponent modelling and calculating odds which are hard to master.

In (Noble, 2002) poker players are evolved by using a fixed topology in sparse networks. These sparse networks consist of only 50 connections, and are evolved by changing weight on connections or changing source or destination of connections. Noble (2002) states that it is necessary to start out by defining some strategies in the population in order for his implementation to evolve strong strategies.

Ravn and Hansen (2005) use NEAT to evolve poker players. Their strategies are based on information about the game that even novice players could deduce from the cards, the table and the opponent's actions. They evolve players playing heads-up poker, which means that the strategy only has to focus on one opponent. Their experiments show that it is not necessary to implement any knowledge about the game in the population.

In this article, the NEAT method will be applied for evolving poker players for the poker variant Texas Hold'em. The players will be seated at tables with 5 opponents, and will use more sophisticated information in their decision-making.

2 NEAT

NeuroEvolution of Augmenting Topologies (NEAT) is a technique of evolving neural network solutions with a minimal topology. This is done by starting out with no hidden layer and then adjust weights while adding hidden neurons and further connections. Only if the strategy with the more complex topology is justified, it can remain in the population.

This is done by speciation of the population and protection of the offspring, so it is not removed prematurely. Furthermore this approach doesn't require any predefined topology, which often is based on experience and experiments.

In NEAT, the population is divided into species based on topological similarities. The distance between two networks is measured as a simple linear combination of the number of excess (E) and disjoint (D) genes, as well as the average weight differences of matching genes (W) .

$$\delta=\frac{c_1 E}{N}+\frac{c_2 D}{N}+c_3 * W$$

Figure 1: The linear combination used to determine the distance between 2 individuals

The coefficients c_1 , c_2 and c_3 are used to adjust the importance of the 3 factors and the factor N , the number of genes in the larger genome, normalizes for genome size. If the distance (δ) between a genome and a randomly chosen member of a species is less than the compatibility threshold, the genome is placed into this species. The values in table 1 are used during the experiment and are found through trial and error, based on the demand of having approximately 15-20 species in each population leading to an average species size of 6.

Table 1: The values used in experiments

Compatibility threshold	1.5
c_1	0.6
c_2	0.6
c_3	0.5

NEAT protects offspring by using *explicit fitness sharing*, which means that a given species' fitness is calculated from its genomes fitness. This must be done because an offspring's fitness cannot be expected to be as good as its parents' fitness, and therefore the offspring could be removed from the population prematurely. Furthermore *explicit fitness sharing* ensures that a given species cannot afford to become too huge, since a species contributes with new offspring based on its shared fitness. This prevents one species from taking over the entire population.

Stanley and Miikkulainen (2002) show that NEAT is performing well at solving the *double pole balancing with velocity information* (DPV) problem, while finding more minimal solutions than other techniques. The principle of starting out with no topology however goes against starting with some predefined knowledge as done in (Noble, 2002). In the experiment presented in this article, some simple knowledge was implemented in the start population by adding connections from some of the input neurons to the output neurons.

3 Method

Texas Hold'em consists of two stages, the preflop and the postflop[1]. Therefore the use of two distinct networks, each covering one of these parts of the problem domain is initiated. This is done, because the evolving of a single network handling both stages is assumed to favour the first stage. If a strategy decides to fold during preflop, the postflop part of the strategy is not utilized, hence, the information from the postflop part to the genetic algorithm will be limited. Furthermore 5 networks are used to handle the opponent modelling. Each of those handles a specific number of opponents, so if the player is facing 3 opponents, the network handling 3 opponents is used. First the preflop or postflop network decides which action to perform. This action is then fed to the proper opponent modelling network, which – if needed - modifies this action accordingly to the opponents' way of playing.

The preflop network consists of 34 input neurons, the postflop network consists of 78 input neurons and the opponent modelling networks consist of 55, 70, 85, 100 and 115 input neurons respectively. All networks have 3 output neurons, fold, check/call and bet/raise. Depending on the stage of the current game, information is fed to the proper network, and the network is activated. The decision of the network is then decided by the output neuron with the highest activity level.

The information available to the network is either Boolean or continuous. The continuous are squeezed with the function $y=\frac{x}{1+|x|}$, in order to be in the interval]-1;1[. At first, the input was normalized using the function $y=x*2-1$, but since the number of inputs set during decision is outnumbered by the number of inputs not set, this normalization blurred the inputs set, since all not set inputs was normalized to -1. This caused the strategies to take unset inputs into consideration, which is why the inputs are not normalized. The information tries to cover all aspects of the game, from information about the hole cards held by the player, to information on the pot odds and how many better hands can be made from the community

[1] Knowledge of the rules of Texas Hold'em is assumed

cards, to actual opponent modelling, where the opponents are considered to be either loose/aggressive, tight/aggressive, loose/passive or tight/passive. Loose/tight refers to the number of hands played by the opponent. The opponent is loose if he participates in many hands and tight if he carefully selects those hands, he's playing. Passive/aggressive refers to the number of raises. If an opponent raises, given the option, most of the times, he's said to be aggressive, whereas an opponent just calling in those situations is passive.

To improve the decision making, information on all, latter half and latter 10% of the hands played by each opponent are fed to the proper opponent modelling network.

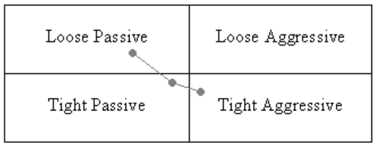


Figure 2: An opponent's table image. The dot to the left indicates all hands seen, the middle dot indicates the latter half and the dot to the right indicates the latter 10% of hands seen. The opponent is changing his style of play. Generally he's playing loose/passive, but recently he's been playing tight/aggressive.

Figure 2 shows that a player can change his playing style over time and that, if the opponent modelling only keeps track of the entire set of hands seen, opponents can change playing style without any major changes in the general playing style.

Information on the opponents is maintained during the game, so as time goes, more accurate information is available. The amount of information is too complicated for most human players to handle at once, but hopefully the results will indicate, which information is most important when playing poker.

The population sizes used during evolution was 108 for all strategies. These populations where kept at a so-called Delegator, which handles the distribution of subpopulations and the genetic algorithm. In each generation step, each population is divided into subpopulations, which is sent to different environments to be evaluated. These environments challenge strategies in different ways. 5 environment types are used; merged, preflop, postflop, model and benchmark. In the merged environment, 18 complete players are evaluated through games against each other at 3 tables. The preflop, postflop and model environments evaluate the preflop, postflop and model part of a player respectively each taking a subpopulation of 36 strategies. The benchmark environment indicates improvements in the populations through games against 3 hand coded strategies; a tight, a loose and a tightbluffer. 4 merged, 1 preflop, 1 postflop, 1 of each model and 3 benchmark environments; one inhabited by tight players, one inhabited by loose players and one inhabited by tightbluff players was initiated during the experiment. The number of environments determines the number of individuals in each population, since each strategy should be evaluated in each generation. Each environment is assured to play at least 600 hands, but instead of waiting for the slower environments to complete their hands, another 120 hands is carried out. Once all environments have played at least 600 hands, the results are returned to the Delegator, where the next generation step is performed.

As in (Noble, 2002) the strategies' winnings were stored in a matrix during the games. A strategy A is said to dominate another strategy B if A's performance against all the strategies in the subpopulation is at least as good as B's performance against all the strategies in the subpopulation and better against at least one. The number of strategies dominated by a given strategy was used to indicate that given strategy's fitness. The 3 most dominant strategies from each merged environment played against a hall of fame of older dominant strategies from previous generations. Only the strategies placing top 3 against all groups of older dominant strategies were added to the hall of fame.

4 Results

The experiment was carried out on 15 P4 3 GHz machines and ran for 14 days resulting in 101 generations. During the experiment 9 randomly selected strategies was sent to all 3 benchmark environments to measure which type of players the populations are performing best against. The results showed that more of the strategies performed better against all three types of benchmark players, which confirms that the genetic algorithm develops robust strategies.

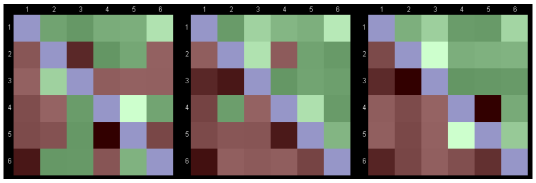


Figure 3: Results from the tight environment. The 3 matrices is from generation 26, 51 and 101. Players from the populations are placed as 1, 2 and 3, and the tight benchmark players are placed as 4, 5 and 6. Blue indicates no development, which only appears in the diagonal. Green indicates improvements from generation 1 and red indicates aggravations from generation 1. The matrices are read by starting at the row for a given player. The colour in a given column

indicates how the row player performed against the column player.

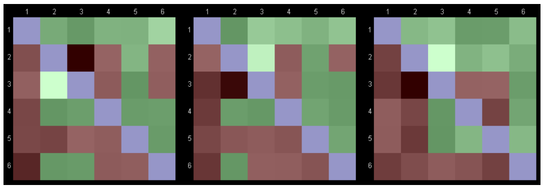


Figure 4: Similar results from the tightbluff environment. Player 1, 2 and 3 are the same as in figure 3.

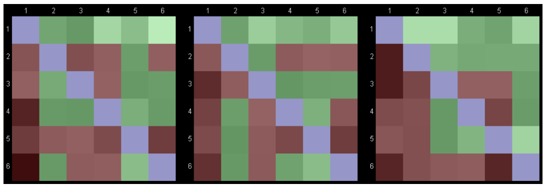


Figure 5: Similar results from the loose environment. Player 1, 2 and 3 are the same as in figure 3 and 4.

Figure 3 shows how 3 randomly selected players from generation 26, 51 and 101 respectively are performing against the tight benchmark players. Compared to the results in figure 4 and 5, one can conclude that player 1 and 2 in generation 101 is robust, since these players perform well against all others opponents in all benchmark environments. In the tight environment all 3 players from the populations are doing better against the benchmark players, which indicates, that the populations abilities to play against tight opponents have increased. It seems however that player 3 in generation 101 is performing worse against the tight bluffers and loose players. This suggests that this player is specialized against tight players.

By examining how often new dominants are found, it can be stated how fast the algorithm finds stronger strategies.

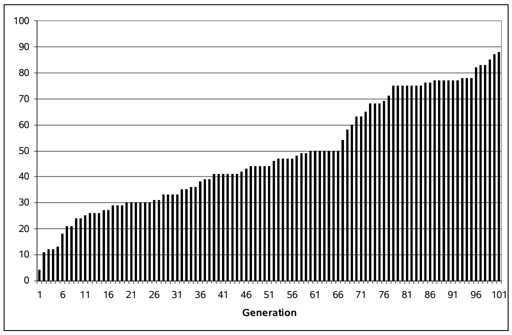


Figure 6: How many dominants are present in each generation.

Figure 6 shows that the number of dominants increases in clusters. This behaviour was also observed in Ravn and Hansen (2005). This suggests that the dominants occur once the algorithm finds an augmenting topology with reasonable weight adjustments and that this solution is improved during the following generations, leading to more dominants. After a while, the algorithm cannot improve the solutions any further without altering the topology, which needs to be adjusted before new dominants are found.

This is also supported by looking at the average number of hidden neurons in the populations and the dominants.

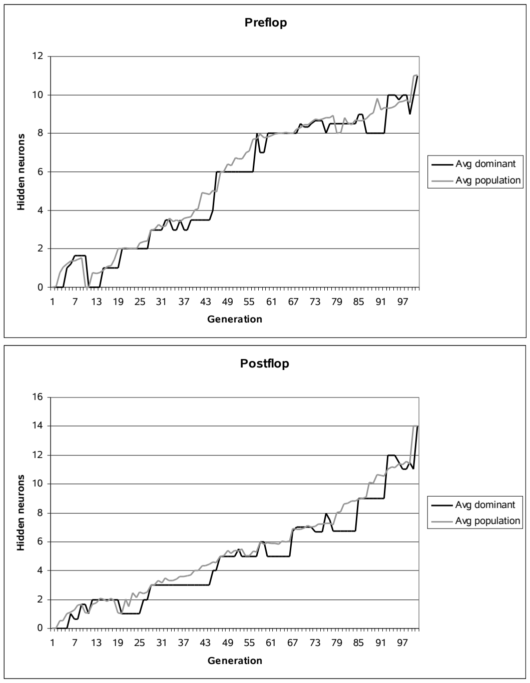


Figure 7: The average number of hidden neurons in the dominant strategies and in the strategies in the preflop and postflop populations.

Figure 7 shows that the dominants have a lesser complex topology than the populations in general. Once a dominant is found, the number of hidden neurons in the new dominants is not increased before the more complex topology is properly adjusted.

These 101 generations are enough to show that the distributed algorithm finds more complex solutions and that the future solutions are stronger. The stronger strategies utilize more advanced knowledge than introduced in the start strategies. This motivates for further generations to be commenced.

5 Future work

This approach can easily be extended to incorporate more environments and it would be interesting to examine the efficiency of introducing more environments. Some environments are likely to contribute more to the evolution, and locating those would also be an interesting aspect to cover. Currently the individuals are evaluated against each other, which require the population to offer the necessary resistance for the individuals to evolve. For instance implementing some of the rules described by Sklansky and Malmuth (1999) and use those to challenge the population would provide increased learning rates.

Another interesting thing would be to replace the NEAT technique with a real-time NEAT technique. This has been used to improve performance for agents interacting in hostile environments by Stanley, Bryant and Miikkulainen (2005). By replacing a poor performing individual with an offspring created from 2 well performing individuals every once in a while, the population evolves during the game in real-time. Human interaction contributes to the abilities of the population, and one can take the role as a trainer and construct different environments in order to stimulate the population during evolution.

Acknowledgements

Special thanks are due to Brian Mayoh for his interesting points of view and feedback.

References

Kenneth O. Stanley and Risto Miikkulainen. Continual Coevolution through Complexification. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2002)*. San Francisco, CA: Morgan Kaufmann, 2002.

Kenneth O. Stanley and Risto Miikkulainen. Evolving Neural Networks through Augmenting Topologies. *Evolutionary Computation 10(2)*. MIT Press, 2002.

Jason Noble. Finding robust Texas Hold'em poker strategies using Pareto coevolution and deterministic crowding. *Proceedings of the 2002 International Conference on Machine Learning and Applications (ICMLA'02)*, CSREA Press, 2002.

Mathias Ravn and Lars Vadstrup Hansen. Neuroevolution of artificial poker players without preencoded expert strategy. *Master thesis in Danish*, Daimi, 2005.

David Sklansky and Mason Malmuth. Hold'em Poker for Advanced Players: 21st century edition. Two Plus Two Publishing, 1999.

Kenneth O. Stanley, Bobby D. Bryant and Risto Miikkulainen.Real-Time Neuroevolution in the NERO Video Game. *IEEE Transactions on Evolutionary Computation (Special Issue on Evolutionary Computation and Games)*. Vol. 9, No. 6, December 2005.

Elastic Systems: Another View at Complexity

Heikki Hyötyniemi*

* Helsinki University of Technology
Control Engineering Laboratory
P.O. Box 5500, FIN-02015 TKK, Finland

Abstract

Elastic systems study complex systems in a cybernetic perspective, assuming that it is the internal feedbacks and interactions that are responsible for the observed functionalities. What is more, in the *neocybernetic* framework there are some additional assumptions — like *balance* and *linearity pursuit* — that make it possible to apply efficient theoretical tools for model construction. It turns out that the very simple starting points are enough to assure *self-regulation* and *self-organization.* This self-organization can be interpreted in terms of multivariate statistics: The system implements *principal subspace compression* of the input resources. This optimality makes it possible to make evolutionary hypotheses, and generalize the models to different phenospheres.

1 Introduction

Stephen Wolfram (Wolfram, 2002) has said it most explicitly: The era of traditional science is over, mathematics cannot give added value to analyses of real-life complex systems. Do we have to be content with weaker descriptions of nature, regressing to "postmodern" ways of doing science (Horgan, 1997)?

Whereas there is motivation for such claims — analyzability truly collapses in powerful enough systems — this is not necessarily a deadlock. For example, Wolfram's *cellular automata* are just another model, and the problems he discusses are problems of that model family.

What if real systems are not fundamentally that complex? Indeed, *neocybernetics* is an approach to complex systems where it is explicitly assumed that modeling still *is* possible. Whereas there are typically many approaches to constructing models, this neocybernetic starting point gives strong guidelines in which directions the modeling has to proceed. There are various assumptions that are somewhat counterintuitive and in contrast with the mainstream complexity theories (these are explained in depth in (Hyötyniemi, 2006)):

- **Universality pursuit.** To make the models scalable beyond "toy worlds", the model structure has to be *linear.*
- **Emergence support.** To make it possible for new structures to emerge, environment-orientedness and *emphasis on data* is needed.
- **Statistical relevance.** To make the observation data representative, the models have to be stationary and *stable* (in the large).
- **Practical plausibility.** The models have to be based on local operations only, as there exist *no structures of centralized control* in nature.

Complexity intuition says that to find interesting behaviors, the models *must* be nonlinear, and they *must not* be in any equilibrium. Similar intuitions apply also to traditional cybernetics (Wiener, 1948). However, in neocybernetics the loss in expressional power is compensated by high dimensionality and dynamic nature of the models. And the homeostasis assumption[1] does not mean that the system would be dead: It has to be kept in mind that one is studying dynamic equilibria rather than static balances, the key functionalities being provided by internal tensions — interaction structures and feedbacks. The emphasis on statistic phenomena means that general attractors rather than individual processes are concentrated on. Strong conceptual tools are needed to maintain the cumulation of complexity in such models, and to give a firm basis for studying *emergence.* Such tools are available in mathematics: Linear algebra with matrix calculus, and system theory with control engineering.

What is the relevance of such discussions to artificial intelligence? It needs to be kept in mind that

[1] Of all possible models, unstable ones are much more common. Neocybernetics does not try to cover all mathematically possible models, only the physically meaningful ones — such that have *survived,* being capable of reaching balance in their environments

cybernetics was one of the cornerstones of early AI, even though the methodologies soon grew apart. It seems that in such mathematical framework some of the age-old AI dilemmas can be attacked: One can find a connection between numeric data and emergent symbolic structures. It can be claimed that *qualitative constructs are attractors of quantitative dynamic processes.*

2 Interpretations of models

Mathematics is a strong language, and it can efficiently be used for "discussing" about new concepts and intuitions. As an example of this, it is shown how the same model can be seen in different ways.

2.1 Towards elastic view of systems

Coupling variables together by introducing a set of constraints is a very general way to define a (linear) model:

$$0 = \Gamma z. \tag{1}$$

Assume that the variables in z are divided in two parts: Vector u, dimension m, describes the environmental conditions, whereas vector $\bar{x}$, dimension n, contains the system-specific *internal variables*, somehow characterizing the equilibrium state of the system. Rewriting the constraints characterizing the system, one can distinguish between the variables:

$$A\bar{x} = Bu. \tag{2}$$

It is assumed that there are as many constraints here as there are latent variables, so that A is square. Because of environment-orientedness, the internal variables are assumed to be directly determined by the environment, so that there assumedly is a (linear) dependency between $\bar{x}$ and u. Assuming that A is invertible, one can explicitly solve the unique linear function from the environmental variables into the system state:

$$\bar{x} = A^{-1}B\,u, \tag{3}$$

so that one can define an explicit mapping matrix from u to $\bar{x}$ as $\phi^T = A^{-1}B.$. However, the main motivation for the formulation in (2) is that one can formally extend the static model into a dynamic one.

The formula (2) only characterizes the final visible global balance in the system, but one has to remember that it is local operations only that exist — how can such uncoordinated local actions implement the global-level behaviors? Indeed, one needs to extend studies beyond the final balance, and take into account the dynamic behaviors caused by the inbalances.

Formula (2) can be interpreted as a balance of tensions determined by forces $A\bar{x}$ and Bu, caused by the system itself and by the environment, respectively. If the forces are not in balance, there is a drift:

$$\frac{d\,x}{\gamma\,d\tau} = -Ax + Bu. \tag{4}$$

The parameter γ can be used for adjusting the time axis. The first term in (4) introduces *feedback.* The steady state equals that of (3), so that $\lim_{\tau\to\infty} x = \bar{x}$ for constant u. Because of linearity, this steady state is unique, no matter what was the initial state. Using the above construction, the static pattern has been transformed into a dynamic pattern — the observed equivalences are just emergent phenomena reflecting the underlying dynamic equilibrium.

How can such a genuine extension from a static model into a dynamic one be justified? It needs to be observed that there *must* exist such a inner structure beyond the surface. The seemingly static dependencies of the form (1) have to be basically dynamic equilibria systems so that the equality can be restored after disturbances: The local actors, whatever they are, do not know the "big picture", and it is the interactions among the actors that provide for the tensions resulting in the tendency towards balance.

What causes the dynamics? Thinking of the mindless actors in the system, the only reasonable explanation for the distributed behaviors is some kind of *diffusion.* It is some kind of gradients that only are visible at the local scale. So, interpreting (4) as a (negative) gradient, there has to exist an integral — a criterion that is being minimized. By integration with respect to the variable x, it is found that

$$\mathcal{J}(x, u) = \frac{1}{2}x^T A x - x^T B u \tag{5}$$

gives a mathematical "pattern" that characterizes the system in a yet another way in a global scale.

For a moment, assume that vector u denotes *forces* acting in a (discretized) mechanical system, and x denotes the resulting *deformations.* Further, assume that A is interpreted as the *elasticity matrix* and B is *projection matrix* mapping the forces onto the deformation axes. Then, it turns out that (5) is the difference between the *internal* and *external potential energies* stored in the mechanical system. Principle of minimum potential (deformation) energy (*et al*, 2001) states that a structure under pressure ends in minimum of this criterion, trying to exhaust the external force with minimum of internal deformations.

However, the formula (1), and thus (5), can be seen to characterize a wide variety of complex balance systems, not only mechanical ones. This means that in non-mechanical cybernetic systems, the same intuition concerning understanding of mechanical systems can be exploited. It does not matter what is the domain, and what is the physical interpretation of the "forces" u and of the "deformations" $\bar{x}$, the structure of the system behavior remains intact: As the system is "pressed", it yields in a more or less humble manner, but when the pressure is released, the original state is restored. Applying these intuitions, one can generally speak of *elastic systems.*

2.2 Pattern matching

There is yet another perspective to see the global pattern that is beneficial because it introduces explicit feedback between the system and its environment.

What comes to the tensions caused by the cost criterion (5), it is equivalent to another criterion:

$$J(x,u) = \frac{1}{2}\,(u-\varphi x)^T\, W\,(u-\varphi x)\,. \qquad (6)$$

This formulation represents a *pattern matching* problem, where one tries to minimize the difference between the "environmental pattern" u and its reconstruction using *features* φ_i weighted by the variables x_i. The correspondence between the cost criteria is reached when one defines the matrices as

$$\left\{\begin{array}{lcl} A & = & \varphi^T W \varphi \\ B & = & \varphi^T W. \end{array}\right. \qquad (7)$$

Criterion (6) gives another view too see the same gradient-based minimization (4). When (6) is minimized using the steepest descent gradient approach, the continuous-time process implementing this minimization is

$$\frac{d\,x}{\gamma\,d\tau} = \varphi^T W\,(u-\varphi x)\,. \qquad (8)$$

It is the gradients that can only be seen by the local actors in a system. From now on, assume the mathematical gradient formulation (8) captures the essence of the natural dynamics, too. Whereas the matrix ϕ^T implements a mapping from the environmental variables u into the system variables $\bar{x}$, the feature matrix φ can be interpreted as an inverse mapping from the space of x back into the space of u. This can be seen as a feedback loop.

However, such observations above have limited value if the data structures ϕ, φ, and W (or A and B) cannot be determined. To attack this problem, a wider perspective is needed.

3 Role of feedback

The main functionality in a cybernetic system comes from the feedback structures. As shown below, there is no need for some "master mind" to implement such interaction structures.

3.1 Exploiting non-ideality

There are no unidirectional effects in real systems: Information flows cannot exist without physical flows that implement them. When energy is being consumed by the system, this energy is taken from the environment, or environmental "resources" are exhausted. To understand these mechanisms, study the pattern matching process (8): There is $-\varphi x$ defining some kind of negative feedback structure, representing real material flow from the system into the environment, denoting the resources being exhausted. The changed environment becomes

$$\tilde{u} = \underbrace{u}_{\substack{\text{actual}\\ \text{environment}}} - \underbrace{\varphi x}_{\text{feedback}}\,. \qquad (9)$$

The system never sees the original u but only the distorted $\tilde{u}$, where the momentary consumption by the system, or φx, is taken into account. Clearly, as the environment affects the system and the system affects the environment, there exists a dynamic structure; again, one is interested in the final balance after transients:

$$\bar{u} = u - \varphi\bar{x}. \qquad (10)$$

Employing this actual, observable environment, one can redefine the mapping ϕ^T so that

$$\bar{x} = \bar{\phi}^T\bar{u}. \qquad (11)$$

When studying the steady state, there is efficiently an algebraic loop in the system, and this means that this structure has peculiar properties. Multiplying (10) from the right by $\bar{x}^T$, taking expectations, and reordering the terms, one receives

$$\mathrm{E}\{(u-\bar{u})\bar{x}^T\}\mathrm{E}\{\bar{x}\bar{x}^T\}^{-1} = \varphi, \qquad (12)$$

so that, when one defines a quantity for measuring the discrepancy between the undisturbed open-loop environment and the disturbed closed-loop environment,

$$\Delta u = u - \bar{u}, \qquad (13)$$

the expression (10) can be written in the form

$$\Delta u = \mathrm{E}\{\bar{x}\Delta u^T\}^T\mathrm{E}\{\bar{x}\bar{x}^T\}^{-1}\,\bar{x}. \qquad (14)$$

Variables in $\bar{x}$ and Δu are mutually connected, they vary hand in hand, but together representing the same mapping as φ. Indeed, this Δu can be seen as a "loop invariant" that helps to see properties of the feedback loop, and it turns out to offer a way to reach simplified analysis of the signals. Because Δu assumedly linearly dependent of u, one can interpret this variable as the actual input driving the whole loop, so that there exists a mapping Φ^T

$$\bar{x} = \Phi^T \, \Delta u. \tag{15}$$

Assuming that the feedback can implement stabilization, the system will search a balance so that

$$\bar{x} = \Phi^T \varphi \, \bar{x}. \tag{16}$$

To have not only trivial solutions (meaning $\bar{x} \equiv 0$), there must hold

$$\Phi^T \varphi = I_n, \tag{17}$$

so that the feedforward and feedback mappings have to be mutually orthogonal. This is a very stringent constraint, and it essentially determines the properties of the feedforward matrix Φ. It turns out that a symmetric formulation is valid:

$$\bar{x} = \mathrm{E}\{\bar{x}\bar{x}^T\}^{-1}\mathrm{E}\{\bar{x}\Delta u^T\} \, \Delta u. \tag{18}$$

This expression connects $\bar{x}$ and Δu with their statistical properties, making it possible to combine two time scales, or emergent levels.

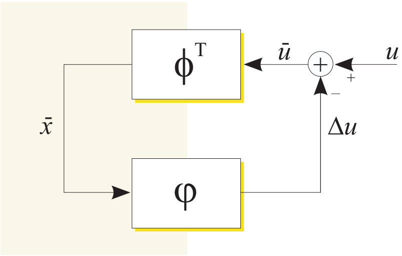


Figure 1: Feedback structure between the system and environment

3.2 Constraints vs. freedoms

Above, the balances of x were studied as the environment u was assumed fixed. However, to reach interesting results, the neocybernetic principles need to be exploited again: It is assumed that there exist various levels of seeing the system, and at each of the levels, the balances are exploited. Whereas u was assumed to remain constant this far, it only has much slower dynamics than x, and on the wider scale, the environment changes. But assuming stationarity of the environment, or balance on the higher scale, so that u has fixed statistical properties, one can find a "balance model of balances". A truly cybernetic model is a *second-order balance model*, or a *higher-order balance model* over the variations in the system — at these levels beyond the trivial first level balance, one can reach stronger views to see the systems, including *self-organization*, as shown below.

So, assume that dynamics of u is essentially slower than that of x and study the statistical properties over the range of $\bar{x}$, and, specially, construct the covariance matrix of it. One can show that there holds

$$\left(\Phi^T \mathrm{E}\left\{\Delta u \Delta u^T\right\} \Phi\right)^3 = \Phi^T \mathrm{E}\left\{\Delta u \Delta u^T\right\}^3 \Phi. \tag{19}$$

If $n = m$, any orthogonal matrix $\Phi^T = \Phi^{-1}$ will do; however, if $n < m$, so that x is lower-dimensional than u, the solution to the above expression is nontrivial: It turns our that *any subset of the principal component axes of the data* Δu *can be selected to constitute* Φ, that is, the columns Φ_i can be any n of the m covariance matrix eigenvectors θ_j of this data. Further, these basis vectors can be mixed, so that $\Phi = \theta D$, where D is any orthogonal $n \times n$ matrix, so that $D^T = D^{-1}$.

These results show that any set of covariance matrix eigenvectors can be selected in Φ. However, in practice it is not whatever combination of vectors θ_j that can be selected: Some solutions are *unstable* when applying the iterative adaptation strategies. Indeed, the only stable and thus relevant solution is such where it is the n most significant eigenvectors (as revealed by the corresponding eigenvalues) that constitute the matrix Φ in convergent systems. This means that the system implements *principal subspace analysis* for input data. Because of the mixing matrix D, the result is not unique in the sense of principal components, but the subspace spanned by them is identical, and exactly the same amount of input data variation is captured.

The properties of principal components are discussed, for example, in (Basilevsky, 1994). It turns out that the directions of *maximum variation* are captured (see Fig. 2). If it is assumed that (co)variation structures in data carry information, it can be claimed that compression of data applying PCA maximally captures the information available.

It turns out that the neocybernetic view of looking at a system is *opposite* to the traditional one: Whereas in (1) one concentrates on *constraints* in the data space, now it is *degrees of freedom* that are concentrated on. This approach is well suited for model-

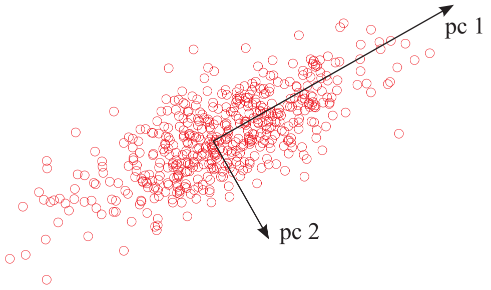


Figure 2: How the PCA basis is oriented towards information — as defined in a technical way

ing of a complex system: There exist assumedly huge numbers of (unknown) feedback structures in a cybernetic system keeping the system in balance; each of these constraints binds system variables closer together, and as a result, there typically remain just a few of the directions in the high-dimensional data space that are non-fixed. The high number of constraints changes into a low number of degrees of freedom — and, applying *Ockham's razor,* a simpler model is *better.*

Now one can conclude that completely local operations result in non-trivial structures that are meaningful on the global scale: Competitive learning without any structural constraints results in self-regulation (balance) and self-organization (in terms of principal subspace). Feedback through the environment, or competition for the resources, results in stabilization and organization of the system.

4 Towards second-order balance

The PCA subspace is determined by the columns φ_i — the above results are of limited use if there is no adaptation in the system. Theodosius Dobzhansky's intuition can be extended beyond biological domains: *No model of a complex system makes sense without reference to evolution.*

4.1 Emergy and evolutionary fitness

The effect of the environmental pressures on an elastic system can be easily quantified: Just as in the case of a potential field, it is the product of the force and displacement that determines the change in potential energy. Similarly, regardless of the physical units of the variables, one can interpret the product $\bar{x}_i\bar{u}_j$ in terms of *energy (power) transferred from the environment into the system* through the pair of variables $\bar{u}_j$ and x_i. This concept deserves a name: In what follows, this "emergent level energy" is studied along the following definition:

> **Emergy** (a scalar dimensionless quantity) is the product of the (abstract) force and the corresponding (abstract) deformation.

As it turns out, this emergy is "information energy" that is the prerequisite for emergence of information structures.

When trying to determine how the system is to adapt, the key challenge is to determine the *goals of evolution.* In natural systems there are no explicit goals as determined from outside: Neocybernetic environment-orientedness suggests a criterion emphasizing some kind of *match with environment.* Indeed, applying the above discussion concerning energy/power transfer from the environment into the system and back, an intuitively appealing fitness criterion would be

> Maximize the average amount of emergy that is being transferred between the system and the environment.

No matter what is the physical manifestation of the environmental variables, a surviving system interprets them as resources, and exploits them as efficiently as possible. However, it must be remembered that there is not only the effect from the external environment into the internal system — there is a symmetric two-way interaction that takes place. It is the matrices $\bar{\phi}^T$ and φ that characterize the emergy transfer between the system and its environment. It is not only so that $\bar{u}$ should be seen as the "force" and $\bar{x}$ as the effect: $\bar{x}$ can be seen as the action and $\bar{u}$ as the reaction just as well.

The momentary emergy traversing from the environmental variable j to the state variable i can be written as $\bar{x}_i\bar{u}_j$, or, when written in a matrix form simultaneously for all variables, $\bar{x}\bar{u}^T$. Similarly, the momentary emergy traversing from the state variable i to the environmental variable j can be written as $\bar{u}_j\bar{x}_i$, or, when written simultaneously for all variables, $\bar{u}\bar{x}^T$. If evolution proceeds in a consistent manner, the differences among such variable pairs should determine the growth rates of the corresponding links between the variables; when the mapping matrices $\bar{\phi}^T$ and φ are defined as shown above, one can assume that a stochastic adaptation process takes place, the observations of prevailing variable levels determining the stochastic gradient direction:

$$\left\{ \begin{array}{lcl} \dfrac{d\,\bar{\phi}^T}{dt} & \propto & \bar{x}(t)\bar{u}^T(t) \\ \dfrac{d\,\varphi}{dt} & \propto & \bar{u}(t)\bar{x}^T(t). \end{array} \right. \tag{20}$$

Because of the feedback through the environment, $\bar{x}$ and $\bar{u}$ remain bounded, and it is reasonable to assume that the processes (20) find a fixed state, or statistical *second-order balance.* The solution for this fixed state can be assumed to be such that the matrix elements $\bar{\phi}_{ji}$ are relative to the correlations between $\bar{x}_i$ and $\bar{u}_j$, or

$$\bar{\phi}^T = q\,\mathrm{E}\left\{\bar{x}\bar{u}^T\right\}, \tag{21}$$

and in the backward direction,

$$\varphi = b\,\mathrm{E}\left\{\bar{u}\bar{x}^T\right\}. \tag{22}$$

This means that the matrices $\bar{\phi}$ and φ should become proportional to each other:

$$\varphi = \frac{b}{q}\,\bar{\phi}. \tag{23}$$

Here, the parameters q and b are *coupling coefficients:* For example, it turns out that q can be interpreted as *stiffness.* As it turns out, these factors scale the signal levels in the system and in the environment.

It needs to be recognized that the adaptation in the system according to (21) is completely local for any element in the matrices $\bar{\phi}$ and φ even though the assumed goal of the evolutionary process is presented in a collective matrix format. As it turns out, the net effect is, however, that the adaptation principle (20) makes φ span the principal subspace of the original input u, and also the subspace determined by Φ becomes aligned.

4.2 Equalization of stiffnesses

The signals $\bar{x}$ and $\bar{u}$, as connected by $\bar{x} = \bar{\phi}^T\bar{u}$, have peculiar properties. For example, multiplying the expression from the right by $\bar{x}^T$ and taking expectation, one has an expression for the latent vector covariance:

$$\mathrm{E}\{\bar{x}\bar{x}^T\} = q\,\mathrm{E}\{\bar{x}\bar{u}^T\}\mathrm{E}\{\bar{x}\bar{u}^T\}^T. \tag{24}$$

This holds *if* the latent variables x_i do not fade away altogether. On the other hand, multiplying the expression from the right by $\bar{u}^T$ and taking expectation, one has

$$\mathrm{E}\{\bar{x}\bar{u}^T\} = q\,\mathrm{E}\{\bar{x}\bar{u}^T\}\mathrm{E}\{\bar{u}\bar{u}^T\}. \tag{25}$$

Substituting this in (24),

$$\mathrm{E}\{\bar{x}\bar{x}^T\} = q^2\,\mathrm{E}\{\bar{x}\bar{u}^T\}\mathrm{E}\{\bar{u}\bar{u}^T\}\mathrm{E}\{\bar{x}\bar{u}^T\}^T, \tag{26}$$

or

$$\frac{1}{q}\,I_n = \bar{\theta}'^T\,\mathrm{E}\{\bar{u}\bar{u}^T\}\,\bar{\theta}', \tag{27}$$

where

$$\bar{\theta}'^T = \sqrt{q}\,\mathrm{E}\{\bar{x}\bar{x}^T\}^{-1/2}\mathrm{E}\{\bar{x}\bar{u}^T\}. \tag{28}$$

From (24), it is evident that there holds

$$\bar{\theta}'^T\bar{\theta}' = I_n. \tag{29}$$

The results (27) and (29) mean that the columns in $\bar{\theta}'$ span the subspace determined by n of the principal components of $\bar{u}$, so that $\bar{\theta}' = \bar{\theta}D$, where $\bar{\theta}$ is a matrix containing n of the covariance matrix eigenvectors, and D is some orthogonal matrix. All eigenvalues $\bar{\lambda}_j$ in the *closed loop* equal $1/q$. This peculiar equalization of system variances is visualized in Fig. 3. When q grows, system *stiffness* increases, and Δu becomes closer to u.

Assume that the coupling coefficients q_i vary between latent variables, so that one has $\bar{\phi}^T = Q\mathrm{E}\{\bar{x}\bar{u}^T\}$ for some diagonal coupling matrix Q. Following the above guidelines, it is easy to see that the matrix of eigenvalues for $\mathrm{E}\{\bar{u}\bar{u}^T\}$ becomes Q^{-1}. What is more interesting, is that one can derive for the symmetric matrix $\mathrm{E}\{\bar{x}\bar{x}^T\}$ two expressions: Simultaneously there holds $\mathrm{E}\{\bar{x}\bar{x}^T\} = Q\mathrm{E}\{\bar{x}\bar{u}^T\}\mathrm{E}\{\bar{x}\bar{u}^T\}^T$ and $\mathrm{E}\{\bar{x}\bar{x}^T\} = \mathrm{E}\{\bar{x}\bar{u}^T\}\mathrm{E}\{\bar{x}\bar{u}^T\}^TQ$. For nontrivial Q this can only hold if latent vector covariance is diagonal; what is more, the vectors in $\bar{\theta}^T = \sqrt{Q}\mathrm{E}\{\bar{x}\bar{x}^T\}^{-1/2}\mathrm{E}\{\bar{x}\bar{u}^T\}$ now not only span the principal subspace, but they are the PCA basis vectors themselves (basis vectors not necessarily ordered in the order of significance). This means that the modes become separated from each other if they are coupled to the environment in different degrees.

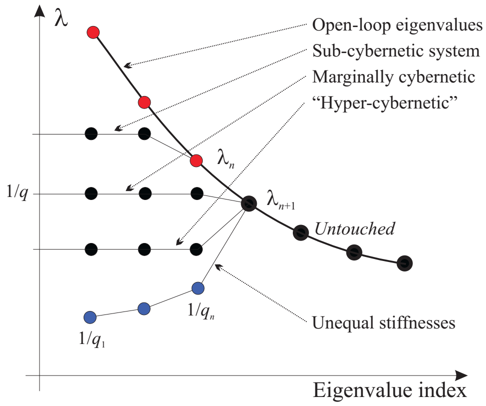


Figure 3: How the open-loop variances are modified by the elastic evolution

5 Elasticity intuitions

Why the above framework was selected, and why the tedious derivations were carried out? As shown below, it can be claimed that the framework of elastic systems makes it possible to reach the age-old goal of complexity theory: Complex systems in very different phenospheres have *the same model.*

5.1 Modeling of populations

The elasticity intuitions and neocybernetic model structures can be applied in artificial intelligence (modeling of signals in infosphere) as well as in artificial life (modeling of signals in biosphere). Perhaps the ideas can also be extended to real intelligence and real life. Here, the AL perspective is first elaborated on.

When modeling some real systems, like populations, it is, in principle, easy to reinterpret the symbols: The vector $\bar{x}$ represents *population sizes*, $\bar{u}$ is the vector of *available resources* (different types of food and other living conditions), and matrices A and B contain the *interaction factors* (competition) among populations. The columns in ϕ can be called *forage profiles*, exploitation conventions corresponding to the populations. But is there such universality among complex systems?

There can exist many alternative ways to survive in the environment — why should one favor one specific model structure? The key point is that *following the neocybernetic model, there is evolutionary advantage.* When the coupling with the environment increases, in a system obeying the neocybernetic model *optimality in terms of resource usage is reached.* In the long run, it is the models that implement the PSA model that can best be matched against variations in the resources $\bar{u}$ (in terms of quadratic variation criteria), resulting in most efficient exploitation of the resources. And populations with optimal strategies assumedly outperform others in terms of biomass and more probable survival. As no other locally controlled model families exhibit the same functionality, *succesfully competing populations assumedly must have adopted neocybernetic strategy.*

It is perhaps hard to believe that the very nonlinear genetic mutations and accomodation processes, etc., would have anything in common with the cellular adaptation details. How could the same model apply? The key observation here is that it is, again, only the dynamic equilibria that are studied, not the all possible routes there. Whereas the adaptation processes can be very complicated and varied, the final emergent optimum can be unique in terms of tensions (see Fig. 4). When concentrating on the balance only, it is also the dimensionality of the problem that goes down, making the analysis feasible.

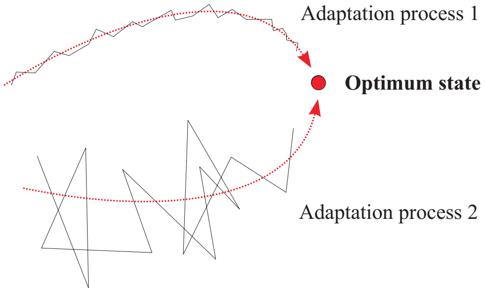


Figure 4: The adaptation strategies and dynamics can be very different in different kinds of systems — but the final state, the hypothetical optimum, is the same for all

Traditionally, ecological models concentrate only on a single species or interactions between two species (for example, see Turchin (2003)). Larger models try to characterize the *niches*, implementing explicit *forage profiles* that describe the resource specifications for each species. However, such models for complete ecologies need careful tuning; evolutionary strategies typically become unstable, meaning that most of the species become extinct, only some of them prospering and exhausting all resources.

When applying the neocybernetic model, ecosystem simulations remain stable even though the dynamics looks “naturally chaotic”: There exists unforced dynamics in different time scales. As the adaptation in the system is based on cybernetic evolution, there is vivid dynamics, but no explosions take place. No fine tuning is needed: If there is enough variation in the resources, after system-wide adaptation a balance is found where there is a “niche” for every species. The niches are characterized by the principal subspace dimensions, the forage profiles ϕ_i mapping from the prevailing resource vector $\bar{u}$ to the balance population $\bar{x}_i$. The roles of the species cannot be predicted, only the subspace that is spanned by all of them together is determined by the environment. The above key observations concerning the neocybernetic model properties can be summarized:

- **Robustness.** In nature, no catastrophic effects typically take place; even key species are substituted if they become extinct, after a somewhat turbulent period. Using the neocybernetic model, this can also be explained in terms of the principal subspace: If the profiles are almost orthogonal, in the spirit of PCA, changes in some of the latent variables are independent of each other, and disturbances do not cumulate. Also because of the principal subspace, the system reacts fast to *relevant* changes in the environment, whereas sensitivity towards random variations that are not supported by the long-term signal properties are suppressed.

- **Biodiversity.** In nature, there are many competing species, none of them becoming extinct; modeling this phenomenon seems to be extremely difficult. Now, again, this results from the principal subspace nature of the model: As long as there exist various degrees of freedom in input, there is room for different populations. Within species, this also explains why in balance there exists variation within populations as the lesser principal components also exist.

Such populations can reside just as well in economies as in ecologies, and the models are applicable for social structures within populations. When relaxing the variables, also *memetic systems* can be studied can be studied applying the same intuitions as trying to do “pattern matching” in the ideasphere. In technical network designs the cybernetic models can help to reach natural-like robustness.

5.2 Power of analogues

When applying linear models, the number of available structures is rather limited – indeed, there exist more systems than there are models. This idea has been applied routinely: Complicated systems are visualized in terms of more familiar systems with the same dynamics. In the presence of modern simulation tools, this kind of lumped parameter simplifications seem somewhat outdated — however, in the case of really complicated, poorly structured distributed parameter systems, such analogies may have reincarnation.

Another class of analogues in addition to the mechanical ones can also be constructed: One can select *electrical current* and *voltage* rather than force and deformation. The external forces change to electrical loads disturbing the system: The deformation is the voltage drop, and the compensating action is the increased current supply (or vice versa).

The electric analogy makes it possible to extend the inner-system discussions onto the environmental level, to inter-system studies. When there are many connected systems interacting, one subsystem exhausting emergy supplied by the other subsystems — or providing emergy for the others, or transferring emergy between them — the criterion for system fitness can be based on the power transmission capabilities among the systems. It is the product of current and voltage that has the unit of power, so that exactly the above discussions apply. Only the intuitions change: Now one can utilize the *inter-system* understanding supplied by electrical systems. Remember that the maximum throughput without “ringing” between electrical systems is reached when there is *impedance matching:* The output impedance in the former system and the input impedance of the latter one should be equal, otherwise not all of the power goes through but bounces back. This same bouncing metaphor can be applied efficiently also in nonelectrical environments — the variables can have different interpretations but the qualitative behaviors remain the same. It is the coupling factors q_i among interacting systems that should be matched by evolution.

6 Conclusion

It can be concluded that elastic systems seem to offer a promising framework for modeling of different kinds of distributed agents. No explicit communication among the agents is needed — higher-level structure emerges when interactions take place through the environment. These claims are best illustrated using practical examples: Such examples of application of elasticity thinking are presented elsewhere in this Proceedings.

References

A. Basilevsky. *Statistical Factor Analysis and Related Methods*. John Wiley & Sons, New York, 1994.

R.D. Cook *et al*. *Concepts and Applications of Finite Element Analysis (4 edition)*. John Wiley & sons, 2001.

J. Horgan. *The End of Science: Facing the Limits of Knowledge in the Twilight of the Scientific Age*. Helix Books, New York, 1997.

H. Hyötyniemi. *Neocybernetics in Biological Systems*. Helsinki University of Technology, Control Engineering Laboratory, 2006.

P. Turchin. *Complex Population Dynamics: A Theoretical/Empirical Synthesis*. Princeton University Press, 2003.

N. Wiener. *Cybernetics: Or Control and Communication in the Animal and the Machine*. John Wiley & Sons, New York, 1948.

S. Wolfram. *A New Kind of Science*. Wolfram Media, Champaign, Illinois, 2002.

Neocybernetic Modeling of a Biological Cell

Olli Haavisto[⋆]
[⋆]Helsinki University of Technology
Control Engineering Laboratory,
P.O.Box 5500, FIN-02015 TKK, Finland
olli.haavisto@tkk.fi

Heikki Hyötyniemi[†]
[†]Helsinki University of Technology
Control Engineering Laboratory,
P.O.Box 5500, FIN-02015 TKK, Finland
heikki.hyotyniemi@tkk.fi

Abstract

A biological cell forms an extremely complex system with a complicated metabolic reaction network. The functionality of the cell is controlled by the genome consisting of thousands of individual genes. Novel measurement technologies provide us with huge amounts of biological data in a quantitative form, but the suitable modeling methods to deal with the high dimensionality and large systems are still missing. In this article, a novel approach for complex system analysis, *neocybernetics*, is applied to modeling a biological cell. It is shown that under certain assumptions both the metabolic network and the gene expression network can be interpreted as a combined linear system. The analysis of this system from the neocybernetic starting points is then elaborated. As an example, the dynamics of yeast gene expression network are modeled according to the presented principles.

1 Introduction

The challenge of complex system analysis is the high number of internal variables and complexity of the connections between them. In the following, we concentrate on a special case of complex systems designated as *elastic systems* (Hyötyniemi, 2006a). A characteristic property of these systems is their ability to adapt to and compensate external disturbances by modifying their internal state. For example, an elastic plate can be deformed by applying external forces to it, and when the forces disappear, the original state of the plate is recovered. It is also assumed that variables in elastic systems are highly connected to each other and there exist internal feedback loops which stabilize the behaviour.

Elastic systems can be analysed using the *neocybernetic* approach proposed in (Hyötyniemi, 2006b), which emphasizes steady states or *balances* of the system instead of the transients leading to these states. It is proposed here that complexity of systems should be attacked by leaving the dynamical transients aside and concentrating on their tendency to remain stable and reach a steady state. Data-based multivariate methods should be utilized to reveal the degrees of freedom of the system variables rather than sticking to the numerous constraints which connect the variables together.

In complex chemical systems elasticity is also manifested: If there are pressures, the system yields. This phenomenon is known as *Le Chatelier principle:* If the environment changes, a new chemical balance is found so that the environmental changes are compensated, at least to some degree. Specially in biochemical systems, this behavior is very dominant, all processes being strongly buffered.

In this article, the vast field of *systems biology* is attacked from a truly systemic viewpoint. Biological cells are interpreted as elastic systems and the usage of neocybernetic modeling principles to model their properties is elaborated. First, a short introduction to cell biology is presented. Then the modeling approach is discussed and a mathematical framework for it is derived. Finally, as an example case, a study of modeling the yeast gene expression dynamics is introduced.

2 Biological cell

The biological cell is the principal unit of all living organisms. Seen from outside, an individual cell can be analyzed as a system that is capable of transferring molecules in and out through its cell membrane. Inside the cell the raw materials are transformed into energy or cell components by a complex machinery involving metabolic pathways, which are strictly controlled by the genetic regulatory network.

Structurally cells can be divided into two main classes, *procaryotes* and *eucaryotes*. Procaryotic cells are typically simpler than eucaryotic ones, since they do not contain membrane separated nucleus and have less intercellular organelles and structures. The cells of one-cellular organisms are usually procaryotic (e.g. bacteria), whereas the multicellular organisms like animals and plants consist of eucaryotic cells.

In the following analysis we are mainly focusing on the properties and functionalities of eucaryotic cells. More specifically, as probably the simplest case of an eucaryotic cell and a typical model organism the brewer's and baker's yeast *Saccharomyces cerevisiae* is considered.

2.1 Genetic regulatory network

Hereditary material of cells is stored in genes, which are parts of the *deoxyribonucleic acid* (DNA) of the *chromosomes* located in the nucleus. Together all the genes form the genome of the organism. A rough estimate of the complexity of an organism is the number of its genes: Simple one-cellular organisms typically contain a few thousand genes, whereas invertebrates have over 10000 and mammals about 30000 genes (see Table 1).

Genes control the cell functions by regulating the amount of proteins inside the cell. When activated, a gene is able to transfer the code it contains into a protein in the process called *gene expression*, which has two parts: *transcription* and *translation*. In transcription, the information stored in the sequence of the nucletides of the gene are read to a *ribonucleic acid* (RNA) molecule, which then moves from the nucleus to the cell organelles called *ribosomes*. There translation takes place: According to the sequence of the RNA molecule, an amino acid chain is constructed. As the chain folds to a stable three dimensional structure, it forms the ready functional protein.

Expression of an individual gene is controlled during all the steps of the expression process. To begin with, transcription is activated only if a correct combination of proteins is attached to the DNA strand. Additionally, by binding to the DNA strand certain gene specific protein complexes called *activators* and *repressors* can either increase or decrease the rate of transcription, respectively. Also the density of the DNA packing, that is, the amount of folding of the DNA double helix affects how easily the gene can be read. After translation, the RNA molecule goes through a series of controlled processes which determine e.g. the final nucleotide sequence and the lifetime of the ready RNA chain. The longer the RNA molecule lasts, the more times it can be used in translation to create proteins. Finally, even the protein folding is a controlled process which determines the usage of the ready protein molecule.

Generally the proteins that control gene expression by binding to the DNA strand are referred to as *transcription factors* and they themselves can be products of other genes. Thus the whole genome forms a connected regulation network with strong feedbacks. As the number of genes is typically high and each transcription factor may affect more than one gene, this network is very complex and hard to analyse.

2.2 Metabolism

The purpose of cell metabolism is to process nutrients, produce energy and keep the cell alive and growing. The complete metabolism of a cell can be divided into many individual and well controlled *metabolic pathways*, which in turn consist of series of metabolic reactions. These reactions are typically catalysed by *enzymes*, a group of proteins produced in gene expression. When a certain gene is activated, more catalysing enzymes for a certain reaction are produced and the rate of the reaction increases resulting to higher concentrations of the corresponding end products. This in turn may affect the activity of some other gene, thus creating a feedback connection between the gene regulation network and the network of metabolic reactions of the cell.

An example of a typical metabolic pathway is shown in Figure 1. In the figure, only the main reactions and metabolites are shown. Each reaction is catalysed by one or more enzymes, shown as boxes with numbers.

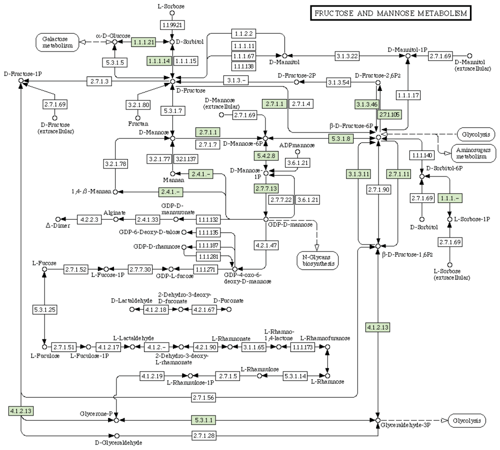


Figure 1: Yeast fructose and mannose metabolism. Each arrow represents a metabolic reaction or a set of reactions catalyzed by an enzyme. Original image from KEGG (www.genome.jp/kegg/)

Table 1: The number of genes of some organisms (Ewing and Green, 2000; Mouse Genome Sequencing Consortium, 2002)

Organism	Group	#genes
Escherichia coli	bacteria	4300
Saccharomyces cerevisiae (yeast)	one-cellular eucaryote	6000
Drosophila melanogaster (fruit fly)	invertebrate	13600
Caenorhabditis elegans (roundworm)	”	19000
mouse	mammal	30000
human	”	35000

For more detailed information on cell biology, see for example the book by Alberts et al. (2002).

3 Modeling approach

Traditionally, investigating the connections and functions in the cells has been concentrated on studying individual genes or chemical reactions for example by disabling a certain gene in some cell cultivation. Behaviour of this cultivation has then be compared to cells having a normal genome. However, using this kind of reductionistic approach leads to difficulties when the combined behaviour of all the small partial reactions and connections should be analyzed, especially as the dimensionality of the problem is large. On the other hand, when using the neocybernetic approach where only the net effects of the whole system are analysed, one is able to form a more holistic picture of the system.

3.1 Dynamics vs. balances

The neocybernetic modeling approach is based on the concept of dynamical balance; it is assumed that the system remains stable and that the system variables are stationary, meaning that their statistical properties remain constant. A biological cell can easily be seen as an neocybernetic system, at least when it is living in a relatively constant environment and remaining healthy and fully functional. In such a case the cell is able to compensate (minor) external disturbances introduced in the form of temperature, pH or chemical concentration variations. When reacting to an external disturbance, the internal state of the cell may change, i.e. balances of the chemical reactions may shift a little and some genes may activate more and some less than before. This means that the cell can adapt to the new environmental situation by changing its internal state and carry on the processes it requires to survive.

The idea of the neocybernetic model is to describe the set of dynamical balances, i.e. the final states of the dynamic system in different homeostatic environments. It is assumed that the inputs of the system (e.g. environmental conditions) are changing slowly when compared with the speed of the internal dynamics of the system. Thus the system state reaches quickly a new balance as the inputs change, and the system mostly stays in balance.

Strictly speaking, biological systems are not in the state of homeostasis, as the living cells continuously exhaust nutrients and produce metabolic products. Here it is assumed that the balanced variables also contain *rates of change,* so that the dissipative processes can also be modeled: It is assumed that in certain conditions the chemical conversion rates remain constant. Inclusion of such derivatives does not ruin the linearity of the model.

3.2 Constraints vs. degrees of freedom

When dealing with dynamical systems containing a large number of variables, the traditional modeling approaches have been based on the *constraints* which determine the relations of the system variables. For example, *differential equations* are a typical choice when modeling the dynamics: Each individual equation covers one relation or constraint between the investigated variables. To unambiguously describe the whole system, as many differential equations are required as there are system variables. However, even in the case of a “simple” biological cell like yeast, for example the gene regulation network includes a huge number of genes (over 6000) which can be connected to each others. Even though the gene regulation networks are typically assumed to be sparse instead of being completely connected, this means that a large number of constraints is required to define the system dynamics. When the system dimension is high, this approach is not feasible anymore.

The opposite way to analyse a multivariable system is to collect data from the system and use them to find out the main directions of variation or *degrees of freedom* present in the system. If the system variables are highly connected and the dynamics are restricted by many constraints, the actual number of meaningful degrees of freedom remains low. Accordingly, it may turn out that the dynamics can be described with a much lower number of variables. That is because each individual constraint reduces the number of possible variation directions by one, thus hopefully resulting to a low dimensional space of degrees of freedom.

3.3 Linearity

Linearity is a strong assumption which is usually not globally valid for real world systems. However, if the analysis is restricted to a small region in the vicinity of a nominal operating point, it may be possible to describe the system behaviour using a linear model. After all, every smooth nonlinearity can be locally approximated to be linear.

There are clear advantages when sticking to linear models: the analyzability and scalability of the models can be preserved even when the model dimension is increased, and there exists a well justified theory for linear systems. On the other hand, if even a minor nonlinearity is allowed, the theoretical analyses become much harder and no general theory exists.

4 Modeling framework

The common factor for different modeling applications in the neocybernetic approach is the assumption of a underlying network structure. Indeed, in both gene expression and metabolism it seems natural to use a network structure as a starting point. In the first case, genes are the nodes of the net, whereas in the latter case the individual molecule concentrations together with environmental factors like temperature and pH value form the nodes. Traditionally when network models are created, e.g. graph theory is used and in order to reduce the complexity of the model, the connectivity of the network is limited. When applying the neocybernetic approach, however, the model is simpler if the network is assumed to be fully connected or *pancausal*. This is because instead of concentrating on the individual connections and their strengths, the aim is to detect the few emerging degrees of freedom.

A linear model structure with multiple variables and parameters can be presented in the form

$$0 = \Gamma^T z, \qquad (1)$$

where column vector z contains the system variables and matrix Γ the parameters. For example, the ordinary d'th order single input, single output (SISO) dynamic system with input u and output y

$$y(k) = \sum_{i=1}^{d} a_i y(k-i) + \sum_{j=0}^{d} b_j u(k-j), \qquad (2)$$

can be defined as

$$\Gamma = \begin{pmatrix} -1 \\ a_1 \\ \vdots \\ a_d \\ b_0 \\ b_1 \\ \vdots \\ b_d \end{pmatrix} \quad \text{and} \quad z(k) = \begin{pmatrix} y(k) \\ y(k-1) \\ \vdots \\ y(k-d) \\ u(k) \\ u(k-1) \\ \vdots \\ u(k-d) \end{pmatrix}. \qquad (3)$$

If there exist more than one constraint between the variables, more columns can be added to matrix Γ and the same structure still remains valid.

In the following it is shown that under certain assumptions both the gene regulation network and the metabolic system of a cell can be written in the form (1).

4.1 Gene regulation network

It has been suggested (Bunde and Havlin, 1994; Barabasi, 2002) that many distributions in self-organized complex networks statistically follow the *power law*

$$z_j = c z_i^D, \qquad (4)$$

where z_i is the free variable, z_j is some emergent phenomenon related to the probability distribution of z_i and c and D are constants. The law (4) can e.g. be applied to relate the popularity of Internet web pages; if z_i is the "ranking of an Internet page" and z_j is the "number of visits per time instant", the dependency of these variables follows the power law.

The single-variable formula (4) can be augmented to include multiple variables:

$$1 = c' z_1^{D_1} \cdot \cdots \cdot z_\mu^{D_\mu}, \qquad (5)$$

where μ is the number of variables. Assuming further that there are multiple dependency structures, the following set of equations is obtained:

$$\left\{\begin{array}{rcl} 1 & = & c_1 z_1^{D_{11}} \cdot \cdots \cdot z_\mu^{D_{1\mu}} \\ & \vdots & \\ 1 & = & c_\nu z_1^{D_{\nu 1}} \cdot \cdots \cdot z_\mu^{D_{\nu\mu}}, \end{array}\right. \tag{6}$$

where ν is the number of dependency structures connecting the variables. Taking a logarithm on both sides of the equations the multiplicative expressions can be transformed to linear structures:

$$\begin{cases} 0 = \log c_1 + D_{11} \log z_1 + \cdots + D_{1\mu} \log z_\mu \\ \vdots \\ 0 = \log c_\nu + D_{\nu 1} \log z_1 + \cdots + D_{\nu\mu} \log z_\mu . \end{cases} \tag{7}$$

Even simpler model structure can be obtained by differentiating the equations around the nominal values $\bar{z}_i$:

$$\left\{\begin{array}{rcl} 0 & = & D_{11} \frac{\Delta z_1}{\bar{z}_1} + \cdots + D_{1\mu} \frac{\Delta z_\mu}{\bar{z}_\mu} \\ & \vdots & \\ 0 & = & D_{\nu 1} \frac{\Delta z_1}{\bar{z}_1} + \cdots + D_{\nu\mu} \frac{\Delta z_\mu}{\bar{z}_\mu}, \end{array}\right. \tag{8}$$

where variables $\frac{\Delta z_i}{\bar{z}_i}$ are relative deviations from the nominal state. Since the original system (7) is nonlinear with respect to the original variables, the differentiated version (8) is only valid locally. However, if we assume only small changes near the nominal values, this model is accurate enough and it is easy to see that it actually is of the form (1).

4.2 Metabolism

The metabolic system of a cell can be characterized as a group of metabolic reactions as explained in Section 2.2. Let us now write one of these equations in the form

$$a_1 A_1 + \cdots + a_\alpha A_\alpha \overset{k_B}{\underset{k_A}{\Leftrightarrow}} b_1 B_1 + \cdots + b_\beta B_\beta, \Delta H, \tag{9}$$

where there are α reactants A_i and β products B_j. Additionally, k_A is the reaction speed in forward direction and k_B in backward direction and ΔH is the change in enthalpy as the reaction proceeds. For each molecule concentration, the rate equation defining the rate of concentration change can be written (see e.g. Atkins (1997)):

$$\frac{dC_{A_1}}{dt} = -k_B C_{A_1}^{a_1} \cdots C_{A_\alpha}^{a_\alpha} + k_A C_{B_1}^{b_1} \cdots C_{B_\beta}^{b_\beta}. \tag{10}$$

In thermodynamical equilibrium all the concentrations remain constant, so that the derivatives are equal to zero. Thus it holds

$$K = \frac{k_B}{k_A} = \frac{C_{B_1}^{b_1} \cdots C_{B_\beta}^{b_\beta}}{C_{A_1}^{a_1} \cdots C_{A_\alpha}^{a_\alpha}}. \tag{11}$$

This equation connects all the system variables together in a multiplicative way. As in the case of gene expression network, the structure can be transformed into a linear one by taking logarithms:

$$\begin{aligned} \log K' = \; & b_1 \log C_{B_1} + b_\beta \log C_{B_\beta} \\ & - a_1 \log C_{A_1} - a_\alpha \log C_{A_\alpha}. \end{aligned} \tag{12}$$

When dealing with local changes in the close vicinity of the nominal operating point $\bar{C}_{B_i}$ and $\bar{C}_{A_j}$, one can further derive the differentiated linear model:

$$\begin{aligned} 0 = \; & b_1 \frac{\Delta C_{B_1}}{\bar{C}_{B_1}} + \cdots + b_\beta \frac{\Delta C_{B_\beta}}{\bar{C}_{B_\beta}} \\ & - a_1 \frac{\Delta C_{A_1}}{\bar{C}_{A_1}} + \cdots + a_\alpha \frac{\Delta C_{A_\alpha}}{\bar{C}_{A_\alpha}}. \end{aligned} \tag{13}$$

In the case of a complete set of metabolic reactions, each reaction leads to its own equation of the form (13). By collecting all the system metabolites into the vector z and adding a column for each reaction in the matrix Γ (having zero coefficients for the metabolites not involved in the reaction), one is able to express the complete metabolic system in the form (1).

4.3 Combining gene expression and metabolism

We have shown that both gene regulation network and metabolic system of a cell can be assumed locally linear phenomena and described using the same framework. Whereas the metabolic reactions proceed quite fast, the gene regulation network has much slower dynamics; it has been estimated that it takes about 15 min for an activated gene to produce enough transcription factors to activate another (target) gene. However, assuming that we only are interested on the steady state behaviour of the system where the transient dynamics have died away, it is possible to combine the two networks. One can simply collect all the gene activation values with the metabolite concentrations and environmental conditions (temperature, pH) into a high-dimensional state vector, and apply multivariate tools to reveal the low-dimensional latent structure containing the degrees of freedom.

4.4 Interpretations of the model structure

When analysing cell functionality it becomes evident that individual components and reactions are not responsible for separate tasks. Instead, each cell function involves multiple reactions and require activity changes in several genes. This means that the behaviour is *redundant* but also *robust*; if one gene or metabolic reaction is disabled, in many cases the cell can overcome the problem and use alternative ways to complete the task. This makes the traditional SISO analysis hard, because strong changes in one variable are required until the effects are seen. However, when analysing all the variables in a neocybernetic manner, the robustness only justifies the assumption of the highly connected network of variables with strong feedbacks.

When the data of the cell are analyzed, some intuitions can be proposed. For example, the degrees of freedom found by multivariate analysis can be interpreted as "functional modes" or cellular functionalities. Each steady state can be described as a linear combination of these directions; some functionalities are going on more than others. It is also possible to analyse the nature of these functional modes; some metabolites and genes are heavily involved in some degree of freedom, thus forming a group of variables connected to that functionality.

5 Example case

As a test case, the gene regulation network of yeast *Saccharomyces cerevisiae* was analysed (for a more technical description of the work, see Haavisto et al. (2006)). The yeast is a common model organism and widely studied because it is easy to cultivate and as an eucaryote contains many functional similarities with more advanced organisms. However, the complexity of the metabolic and genetic systems of yeast is very high, and the cells are capable of exploiting multiple different nutrients and live in varying environments (Walker, 2000).

There are two main types of yeast genome time series experiments available, namely stress experiments and cell cycle measurements. In this study we focused on the former type, where the environmental conditions of a non-synchronized yeast cultivation are disturbed, and a time series of the activities of the genes is measured after the shock. Since there are a large number of cells in the cultivation, other genomic activities like cell cycle are averaged away, and in the data only the effects of the shock are seen. These series are also referred to as *stress experiments*.

It was assumed in the analysis that the stress reaction of yeast cultivation proceeds as follows. Originally, the cultivation is grown in a static environment, so that it reaches some density and the average of the cell internal states remains in the nominal level. At the time instant zero, the environmental conditions are suddenly changed, for example the temperature of the cultivation medium is increased from 25°C to 30°C. As a response to this, the activities of several genes are either increased or decreased as the cells are trying to adapt to the new situation. This *transient phase* ends when the gene activities stabilize to a new steady state, where the levels of the activities typically differ from the nominal state. In this case, it could be defined that only the transient phase is actually 'stress', whereas the steady states are interpreted as normal function states of the cultivation.

5.1 Data

The gene activities of an organism with known genome can be measured using the *microarray* technology (see e.g. Schena (2003)), which gives a "snapshot" of the activities of all the genes at a certain moment. This rather new technology provides huge amounts of biological data and has made possible the data-based analysis of biological cells using the increased computational capacity and data mining tools. However, to model the dynamic behaviour of the genome still remains a nontrivial task, since a typical time series of gene activities contains about 10 time points. As the system dimension e.g. in the case of yeast is over 6000, the problem is highly underdetermined.

In this study two publicly available data sets were utilized, which originally are published by Gasch et al. (2000) and Causton et al. (2001). These stress response experiments both include responses to temperature and pH changes as well as addition of some chemicals (e.g. salt, sorbitol, etc.). Each time series contains measurements of the transient phase, and it is assumed that in the end of the series new balance is reached. There were altogether 21 time series with a total of 152 measurement points. After preprocessing the data and discarding the incomplete genes, there remained about 4000 genes.

5.2 Modeling

As discussed in (Hyötyniemi, 2006b), the framework of elastic systems makes it possible to make hypotheses concerning *goals of evolution.* It can be assumed that individual actors try to maximize their coupling with the environment, simultaneously maximizing the intake of resources. If this is the case in the real cell, and if there has been enough time for the evolutionary mechanisms to find the optimum, strong mathematical intuitions and tools are available. It is not whatever degrees of freedom that are manifested in data, they are axes of *principal components* (see e.g. Basilevsky (1994)). And when such principal component analysis is carried out for the observation data, it is not just some data analysis; it can be claimed that it is *system analysis,* the mathematical structures corresponding to real physiological structures.

Instead of modeling only the degrees of freedom present in the final steady states of the time series, also the dynamics of the genome were included in the model. This violates the assumption of steady states with stationary statistical properties, since during the transient phase additional degrees of freedom activate and thus the latent dimension of the system state may remain high. However, the results obtained also contain the initial and final states where the assumptions hold.

For creating a dynamic model of gene regulation network we utilized a fairly new modeling method, *subspace identification* (Van Overschee and De Moor, 1996). The method is especially suitable for multidimensional system analysis. However, due to the small number of available data points and their high dimensionality, some modifications to the steps of the basic algorithm had to be made. The method produced a discrete state space model with the environmental conditions (pH, temperature and cultivation medium molecule concentrations) as inputs and gene activity levels as outputs. This kind of model is suitable for example in development of a *Kalman filter*, which can optimally estimate the system state (see e.g. Grewal and Andrews (1993)).

In the modeling phase, the system internal dimension was selected to be four based on the singular value inspection during the subspace identification algorithm calculation. As a result, a drastic model complexity reduction takes place when compared to the original number of genes. This means that all the relevant functional modes present in the stress responses can be coded using these four degrees of freedom, and even some of the transient phase variations are allowed.

5.3 Results

The obtained dynamical model could be used to estimate or simulate the yeast gene expression when an environmental change is introduced. To test the quality of the model, responses to the environmental changes present in the original data were simulated by the model, and the results were compared to the measured values. Generally the modeling results were good; the correlation coefficient between the measured and simulated values for individual experiments varied mainly around the value 0.8, even though a couple of lower values were also detected.

Figure 2 shows the measured and simulated responses of the yeast genome to a step addition of salt. On the left, the measured values are shown, whereas on the right are the simulations. Each row of the figure corresponds to one gene, and each column represents one time point. White color corresponds to high and dark color to low activity values. The number of genes was limited for visualization reasons to the group of 255 stress-related genes. However, the model was able to simulate the activities of all the 4000 original genes.

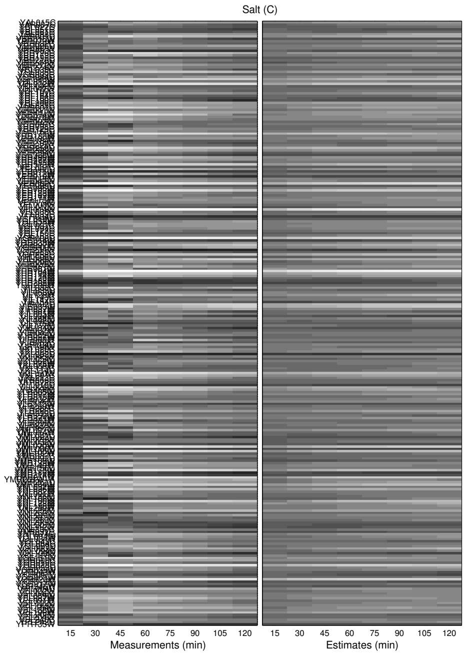


Figure 2: Yeast gene expression response to salt addition, measured and simulated values

When analysing the figure, it can be noted that even though there may be differences between the measured and simulated values on the transient phase, the steady states are modeled quite well. This is because the original neocybernetic assumption of dynamical balance holds only in the steady state, whereas during the transient phase more complex dynamics and additional degrees of freedom are present in the data.

Because of the lack of data, it was not possible to use a proper validation data set. However, since the model dimensionality is strongly reduced when compared with the original system, it can be assumed that the generalization capability of the model should be quite good. That is, the model is actually able to catch some properties of the real phenomena that produced the data instead of just modeling the given data set.

6 Conclusions

A neocybernetic approach for modeling complex systems was discussed in this article. It was shown that both gene expression network and metabolism of a cell can be approximated to follow the proposed linear structure. Utilizing this, data-based principles for analysing the behaviour of cell cultivations were elaborated and a case study of modeling the dynamics of yeast gene expression was presented. The results of the case study encourage the usage of the presented approach for gene expression modeling at least at the high abstraction level.

Acknowledgements

This work was partly done in the project SyMbolic (Systemic Models for Metabolic Dynamics and Regulation of Gene Expression), funded by the NeoBio program of National Technology Agency of Finland (TEKES) during 2004-2006. OH has also been supported by the Technology Promotion Foundation (TES).

References

Bruce Alberts, Alexander Johnson, Julian Lewis, Mrtin Raff, Keith Roberts, and Peter Walter. *Molecular Biology of the Cell, 4th edition*. Garland Science, 2002.

Peter W. Atkins. *Physical Chemistry*. Oxford University Press, 1997.

Albert-Laszlo Barabasi. *Linked: The New Science of Networks*. Perseus Publishing, 2002.

Alexander Basilevsky. *Statistical Factor Analysis and Related Methods*. John Wiley & Sons, New York, 1994.

Armin Bunde and Shlomo Havlin, editors. *Fractals in Science*. Springer Verlag, 1994.

Helen C. Causton, Bing Ren, Sang Seok Koh, Christopher T. Harbison, Elenita Kanin, Ezra G. Jennings, Tong Ihn Lee, Heather L. True, Eric S. Lander, and Richard A. Young. Remodeling of yeast genome expression in response to environmental changes. *Mol Biol Cell*, 12:323–337, 2001.

Brent Ewing and Phil Green. Analysis of expressed sequence tags indicates 35,000 human genes. *Nature Genetics*, 25:232–234, 2000.

Audrey P. Gasch, Paul T. Spellman, Camilla M. Kao, Orna Carmel-Harel, Michael B. Eisen, Gisela Storz, David Botstein, and Patrick O. Brown. Genomic expression programs in the response of yeast cells to environmental changes. *Mol Biol Cell*, 11:4241–4257, 2000.

Mohinder S. Grewal and Angus P. Andrews. *Kalman filtering theory and practice*. Prentice Hall, 1993.

Olli Haavisto, Heikki Hyötyniemi, and Christophe Roos. State space modeling of yeast gene expression dynamics. Submitted to *Journal of Bioinformatics and Computational Biology*, 2006.

Heikki Hyötyniemi. Elastic systems: Another view at complexity. In *Proceedings of SCAI '06*, 2006a.

Heikki Hyötyniemi. Neocybernetics in biological systems. Technical report, Helsinki University of Technology, Control Engineering Laboratory, to be published, 2006b.

Mouse Genome Sequencing Consortium. Initial sequencing and comparative analysis of the mouse genome. *Nature*, 420:520–562, 2002.

Mark Schena. *Microarray Analysis*. Hoboken, NJ, 2003.

Peter Van Overschee and Bart De Moor. *Subspace Identification for Linear Systems*. Kluwer Academic Publisher, Boston, Massachusetts, 1996.

Graeme M. Walker. *Yeast Physiology and Biotechnology*. John Wiley & Sons, 2000.

Application of Elastic Intuitions to Process Engineering

Kalle Halmevaara[*]
[*]Helsinki University of Technology
Control Engineering laboratory
PO Box 5500, 02 015 TKK, Espoo, Finland
kalle.halmevaara@tkk.fi

Heikki Hyötyniemi[*]
[*]Helsinki University of Technology
Control Engineering laboratory
PO Box 5500, 02 015 TKK, Espoo, Finland
heikki.hyotyniemi@tkk.fi

Abstract

In this paper the intuitions of the elastic systems and neocybernetics are applied to large scale industrial systems. The process performance enhancement can be seen as technical evolution of the system. The objectives of technical evolution are somewhat different from those of natural neocybernetic systems. The performance of the system can be characterized by means of quality measures and the parameters of the system can be tuned along the axes of freedoms in the parameter space towards values that result in better process performance. Results from a simulated case study on a continuous pulp digester model are presented and discussed.

1 Introduction

Traditionally, managing complexity in large-scale industrial systems has been tackled with reductionist engineering approaches. One concentrates on the analysis of individual signal realizations – however, it is only different kinds of *quality measures* emerging from statistical considerations that are the most important from the point of view of mastering the process and production. The low-level structures are typically well-known – ideal mixers, etc. – but how are the properties of the high-level structures – networks of such mixers – related to them?

It turns out that thinking of the complex plants as *elastic systems* helps to find the conceptual and practical tools to reach the level of *emergent behaviors* of such systems (Hyötyniemi, 2006). One never knows all relationships among variables in a complex system – but in the elasticity framework this does not matter. It only has to be assumed that the underlying interactions and feedbacks manage to keep the system in balance in varying conditions – and this assumedly is a realistic assumption: Only realistic, successfully operating processes are studied.

Whereas the interactions and feedbacks determine the *constraints,* it is the remaining *freedoms* that are more interesting. The number of degrees of freedom is typically much lower than the total number of system parameters is, so that when concentrating on the emergent functionalities, or the manifestations of the freedoms, low-dimensional models can be found. When the system is “pressed", the system yields along the freedom axes – if the disturbances are small, one can assume that there is a linear relationship between the *tensions* and the corresponding *deformations.*

The industrial plant is very much like a biological cell (Haavisto, 2006): It also tries to maintain its *homeostasis,* and it tries to keep up its “metabolism", or production, regardless of the external disturbances. There are differences, too: “Evolution" in such a system is implemented not by nature but by an “intelligent designer", the human. In a natural system there is no predetermined direction of evolution – adaptation simply takes place in the direction of maximum cross-correlation among signals, in the sense of *principal components* (see Hyötyniemi, 2006); however, in the industrial case explicit goals exist. There is now more structure, as the signals have different roles: There are the input variables, the *qualifiers* that can be affected, and then there are separately the output variables, or *qualities.* The process of adaptation thus becomes more sophisticated: When the outputs are to be maximized, there exist plenty of alternative regression structures (like *canonical correlations*) that can be exploited to modify the directions of the freedom axes to make them better reflect the design goals.

After the degrees of freedom are found, the qualifiers (model or control parameters, set points, etc.) can be gradually modified to “push" the qualities in the desired directions. This way, a “holistic" way of iterative optimization of the complex system can be implemented.

In this paper, a holistic Iterative Regression Tuning (IRT) method applying neocybernetic ideas for tuning multiparameter systems is presented. Originally the IRT method and results from earlier case studies were presented in (Halmevaara and Hyötyniemi, 2004; Halmevaara and Hyötyniemi, 2005). First, the IRT method and some practical instruments are introduced. After that, an application on continuous pulp digester control system is presented. Finally, the results and the potential of the IRT method in other applications are discussed.

2 IRT method

Any dynamic (or static) system with multiple input and output variables, $x = (x_1\ x_2\ \ldots)^T$ and $y = (y_1\ y_2\ \ldots)^T$, respectively, can be described simplistically as,

$$y(t) = G\big(x(t);\theta\big) + e(t), \tag{1}$$

where θ are the system parameters and e stands for the stochastic variations in y. The function G determines the internal structure of the system. For example, G can stand for a large process model including its control system in which case θ correspond to control parameters, setpoint values and other numerical process parameters. The performance of the system can be evaluated by examining the values of the output variables with respect to the applied inputs. The performance can be evaluated in general by means of quality measures q, see Figure 1. E.g., robustness and accuracy are important concepts in describing the performance of a control system. Such concepts can be measured, e.g., with quality measures like variance and setpoint tracking ability of controlled variables.

The values of the m quality measures, $q = (q_1\ q_2\ \ldots\ q_m)^T$, corresponding to certain values of n parameters, $\theta = (\theta_1\ \theta_2\ \ldots\ \theta_n)^T$, are calculated from a set of input and output signals, x and y, using some user defined smooth continuous functions. The signals can be recorded from the existing system or they can be produced using a simulation model of the system in consideration. Since the system outputs y are affected by random variations the quality measures q are also random variables. However, it turns out that there exists a statistical dependency between θ and q, which can be modelled if enough data is available. Starting from some initial values of θ and varying their values randomly in a local domain, a set of data, (Θ,Q), is obtained. Notice that the lower case symbols (e.g., θ and q) are used here for the single samples of (multidimensional) data vectors and the upper case symbols (e.g., Θ and Q) are used for the data matrices such that $Q = (q(1)\ q(2)\ \ldots\ q(k))^T$ and $\Theta = (\theta(1)\ \theta(2)\ \ldots\ \theta(k))^T$, where k is the number of data points. The data acquisition can be done with Markov Chain Monte Carlo simulation, or to minimize k, with a pseudo-random sampling of the parameter space (experiment design).

If the joint distribution of Θ and Q forms a single multivariate Gaussian distribution, the dependency between θ and q is linear in the maximum likelihood sense (see, e.g., Pindyck and Rubinfeld, 1991),

$$\Delta q = F^T \cdot \Delta\theta + \varepsilon, \tag{2}$$

or using the matrix notation,

$$\Delta Q = \Delta\Theta \cdot F + \mathrm{E}. \tag{3}$$

In equations (2) and (3), Δ corresponds to deviation from the current performance $\overline{q}$, i.e., $\Delta q = q - \overline{q}$, and ε and E are the residual errors of the linear model. Notice that in the quality measure calculation the time axis is abstracted away and therefore even the dynamic systems are now examined using static models. Now, θ and q can be interpreted as variables on the slower emergent scale, the dynamics of the original variables x and y being considerably faster.

The performance of a large process and its control system requires several quality measures to be characterized thoroughly. Also the number of the parameters rises along the size of the system. Thus, the size of the linear equation system (2) also grows making the determination of the mapping matrix F nontrivial. If a sufficient amount of data from the system can be obtained the local linear model can be estimated using statistical multivariate methods like Partial Least Squares (PLS) or Canonical Correlation Analysis (CCA) based regression techniques (Hyötyniemi, 2001). Instead of concentrating on the correlations among θ (Principal Component Analysis) the aim is now to capture cross-correlations between θ and q.

The tuning task of the parameters can be formulated as a quadratic optimization problem,

$$J = q^T W q, \tag{4}$$

where J is the cost function to be minimized and W is a diagonal weighting matrix. The problem cannot be solved analytically and nonlinear (numerical) gradient based optimization methods, such as conjugate gradient method (see, e.g., Gill et al., 1981), have to be applied. The cost function can be approximated locally as

$$\begin{aligned}\hat{J} &= \hat{q}^T W \hat{q} \\ &= \big(\Delta\theta^T F + \overline{q}\big) W \big(F^T \Delta\theta + \overline{q}\big),\end{aligned} \tag{5}$$

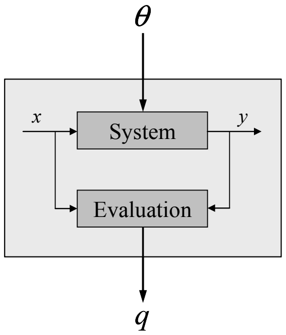


Figure 1: The statistical dependency between parameters, θ, and quality measures, q, describes the relevant properties of the underlying dynamic MIMO system, having inputs u and outputs y (Hyötyniemi, 2002).

which is differentiable with respect to θ. Using the following update principle, the current values of the parameters, $\bar{\theta}$, can be tuned iteratively,

$$\bar{\theta}(K+1)=\bar{\theta}(K)+\gamma d(K), \tag{6}$$

where K is the update step index, γ is the length of the update step and d is the search direction, which is determined by the chosen optimization procedure. E.g., in the basic gradient descent method d is calculated by differentiating (5) with respect to θ, and in the moment method exponential smoothing is applied to combine the successive gradients to obtain a more robust search direction.

The IRT method requires reasonable initial values for θ since local optimization methods are used. When concentrating on control parameter tuning there exists a great number methods to assist in single input single output (SISO) controller tuning (see, e.g., Harris et al., 1999; Thornhill and Hägglund, 1997; Hägglund, 1999). In simulation model tuning the domain area expertise is used to obtain reasonable starting values for the model parameters. The IRT procedure requires also some steering, e.g., with respect to range of the local variation of θ, appropriate selection of γ, and number of data points suitable for obtaining the local model F. To obtain somewhat automated version of the same procedure the following enhancements are proposed.

2.1 Gaussianity testing

Based on domain area expertise the level of the local variation of θ is specified. To enhance the signal-to-noise ratio it is desirable to increase the local variation, but on the contrary, the local linearity between θ and q is lost if too large area of the parameter space is studied in one iteration step. Therefore, some means for Gaussianity testing on q are required.

When selecting an appropriate test one has to consider the power of the test, i.e., what is the probability that the incorrect null hypothesis on Gaussianity is rejected. Second, the distribution of the test statistic or at least some critical points of it should be known. And third, the hypothesis testing can be done using either known or unknown distribution as an alternative hypothesis. If the alternative distribution (the form of non-Gaussianity) is unknown, an omnibus test has to be chosen (like in this case).

First it is assumed that it is sufficient to consider the marginal distributions of the multinormal distribution which give the necessary (but not sufficient) conditions for the multivariate normality. This simplification can be argued with the fact that the multivariate normality tests are computationally fairly intensive, and their sensitivity to different types of departures from the multinormality vary significantly, i.e., it has been recommended to use more than one test especially when the characteristics of the non-normality are unknown a priori (Thode, 2002). Further, since many of the multivariate tests are extensions of the corresponding univariate tests, they require the calculation of the order statistics, which cannot be done explicitly in the multivariate case.

Three most powerful univariate omnibus tests are Wilk-Shapiro, Anderson-Darling and the joint kurtosis-skewness tests (Thode, 2002). The Wilk-Shapiro test is the most attractive since the distribution of the test statistic is known. Also, if the improvements suggested by Royston (Royston, 1982) are applied, the Wilk-Shapiro test is valid for sample sizes $7 \le k \le 2000$. The test compares the observations to the theoretical order statistics of the Gaussian distribution.

The Gaussianity testing also serves as an outlier detection method, since the tests are typically sensitive to strongly differing samples. It can also point out the improper quality measure definitions.

2.2 Reliability of the local linear model

No matter how well the linearity assumption is satisfied, the tuning procedure always returns some search direction at every iteration step. Since the local linearity assumption may have been violated severely, the reliability of the following update step in that direction should be somehow taken into account. One way to accomplish this is to study the confidence intervals of the estimated parameters F of the linear model which gives at the same time confidence intervals for the gradient vector. This can be done, e.g., using Bayesian analysis of the regression model coefficients (Gelman et al., 2004). The posterior joint distribution of the model parameters F and the residual variance $\sigma_\varepsilon{}^2$,

$$p\left(F,\sigma_\varepsilon^2 \middle| q,\theta\right) = p\left(F \middle| \sigma_\varepsilon^2,q,\theta\right) p\left(\sigma_\varepsilon^2 \middle| q,\theta\right), \quad (7)$$

can be utilized to obtain confidence intervals for the estimated linear model. In (7) the first term on the right is the conditional posterior distribution of F, given $\sigma_\varepsilon{}^2$, and the second term is the marginal posterior distribution of $\sigma_\varepsilon{}^2$. The confidence intervals can assist to define a suitable length for the update step, γ. If it is uncertain that the update step in the search direction results in an improvement, the value of γ should be reduced. Further, wide confidence intervals suggest that more data should be applied in the parameter estimation to obtain more reliable prediction.

2.3 Iterative PLS regression

In a data-based iterative tuning of θ (6) the local linear model (2) is used in the approximation of the local gradient to determine d. Usually, if n and m are reasonably small, it is sufficient to solve F using standard multilinear regression (MLR). In many cases, however, MLR fails due to collinearities and noise in the multidimensional data. These numerical problems can be avoided using more advanced multivariate regression methods.

The PLS regression model can be defined using an eigenproblem-oriented framework (Hyötyniemi, 2001). The resulting PLS latent basis coincides with the traditional algorithmic definition of PLS only with respect to the most significant basis vector. However, the method provides one with means for iterative PLS regression.

The starting point in finding a good latent variable basis is to maximize the correlation between the x and y. Assuming that the input data X is projected onto a latent basis φ, and the output data Y is projected onto basis ϕ, the objective is to find such bases that the correlations between the projected variables are maximized. To obtain a unique solution the lengths of the basis vectors have to be restricted somehow. This results in a constrained optimization problem,

$$\begin{cases} f\left(\varphi_i,\phi_i\right) = \frac{1}{k}\cdot\varphi_i^T X^T \cdot Y\phi_i \\ g_1\left(\varphi_i\right) = 1-\varphi_i^T\varphi_i \\ g_2\left(\phi_i\right) = 1-\phi_i^T\phi_i \end{cases} \quad (8)$$

where the subscript i refers to a single latent basis vector. Using the method of Lagrangian multipliers to solve (8) results in a pair of eigenvalue problems,

$$\begin{cases} \frac{1}{k^2}X^T YY^T X\varphi_i = \lambda_i\varphi_i \\ \frac{1}{k^2}Y^T XX^T Y\phi_i = \lambda_i\phi_i \end{cases} \quad (9)$$

where λ_i is the eigenvalue corresponding to the eigenvectors φ_i and ϕ_i. It is sufficient to solve one of the equations in (9). The eigenvalue decomposition of the cross covariance matrix of the projected data gives the complete set of latent basis vectors from which the N most significant can be chosen. E.g., if the first equation in (9) is utilized, the calculation of the covariance matrix can be expressed in an iterative form using exponential smoothing,

$$R\left(K+1\right) = \mu R\left(K\right) + \left(1-\mu\right)X^T YY^T X, \quad (10)$$

where $\mu \in (0, 1)$ is the forgetting factor and R is the cross covariance matrix. Using the iteratively updated R in the determination of the PLS latent basis, one gets rid of the zigzag effects plaguing the standard gradient descent algorithm. This can be seen as an alternative optimization method to moment method for example. Now the whole covariance structure of the data is preserved from one iteration step to another instead of only the gradient vector.

3 Pulp digester case study

A model of a Finnish pulp mill was applied in the test case. The process model was constructed with the APROS software that is a professional simulation environment for modeling combustion and nuclear power plants, and pulp and paper mills (see, e.g., Juslin and Paljakka, 1997). APROS provides large libraries of process and automation components, which can be combined into rigorous models of industrial plants. APROS has been used successfully, e.g., in various training simulator and process analysis projects. In the following, the model of the pulp mill is introduced, after which the tuning targets of the case study are formulated.

3.1 Model and problem description

A general overview of the model is presented in Figure 2. First, the woodchips and the impregnation liquor are mixed, after which the mixture is fed into the impregnation vessel. The flow continues to the top of the digester, where the mixture is heated with steam to the cooking temperature. The chips were modelled to consist of several cellulose, carbohydrate and lignin components, and the liquor was assumed to contain sodium hydroxide and sodium sulphide in addition to the organic compounds dissolved from the woodchips. Chemical reactions during the cooking phase were modelled according to (Gustafson et al. 1983). Several circulation streams of black liquor to the preceding parts of the process were modelled. The subsequent washing and bleaching operations were excluded from the model.

In the case study, the production of pulp in a steady state operation was considered, i.e., changes of neither production rate nor quality targets were simulated. Initially, several control loops were behaving poorly. The level controllers in the impregnation and the digester vessels were oscillating heavily due to their inappropriate tuning. Consequently, also the washing coefficient control failed to meet its targets. The two model-based controllers were behaving even more detrimentally. The other one predicts the kappa number of the produced pulp based on the digester top temperature measurement. The five hour time delay in the cooking process makes the prediction essential for the process control. The other model calculates the setpoint value for the H factor based on the measured temperature profile of the digester, the amount of applied alkali and the kappa number of the produced pulp. Both models were giving strongly biased predictions, causing naturally serious problems. Due to the improper controller tuning the process was producing continuously out of specification pulp.

3.2 Optimization targets

The optimization cost function was formed with six quality measures. The six considered variables were:

y_1: kappa number in the digester blow,
y_2: washing coefficient,
y_3: liquor level in the digester,
y_4: chip level in the digester,
y_5: chip level in the impregnation vessel, and
y_6: H factor prediction.

The control error of the variables y_1 - y_5 and the prediction error of y_6 was tried to minimize. The tuning involved the parameters of seven PI controllers and two models that were applied in prediction and setpoint calculation, i.e., the total number of parameters was n = 20. The PI controllers were responsible for regulation of the chip and liquor levels, washing coefficient, production rate, H factor, and the digester steam chamber temperature.

Here are presented the results of K = 30 iteration steps. The fairly great number of global steps is due to the applied gradient method. Also, since the initial performance was purposefully set to extremely poor, long tuning time was understandable. In the case study k = 50 data points on the average were simulated on each iteration step and the length of a simulation run was T = 8h. Simulations were run about 25 times faster than real time.

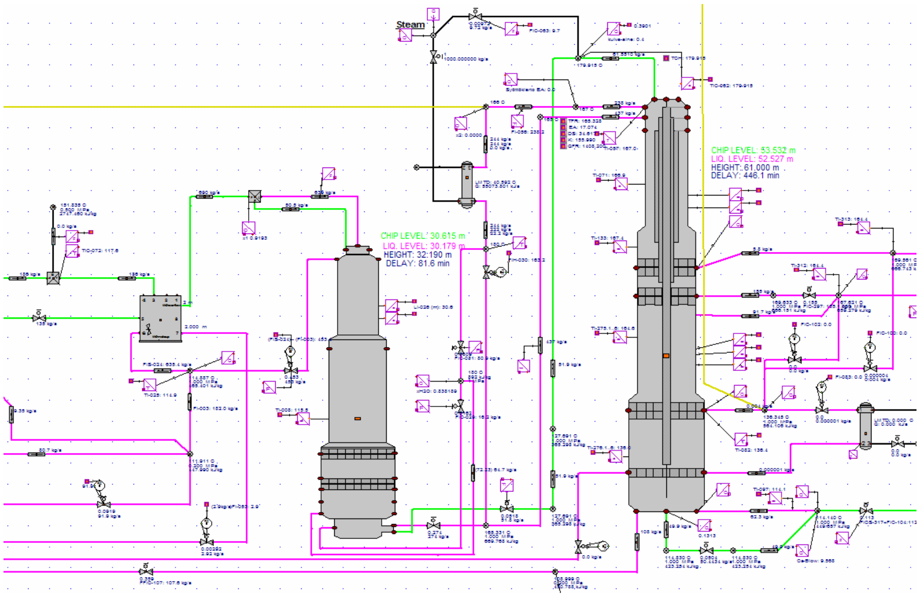


Figure 2: The model of the pulp mill: The chip feed screw conveyor (left), the impregnation vessel (in the middle), and the pulp digester (right).

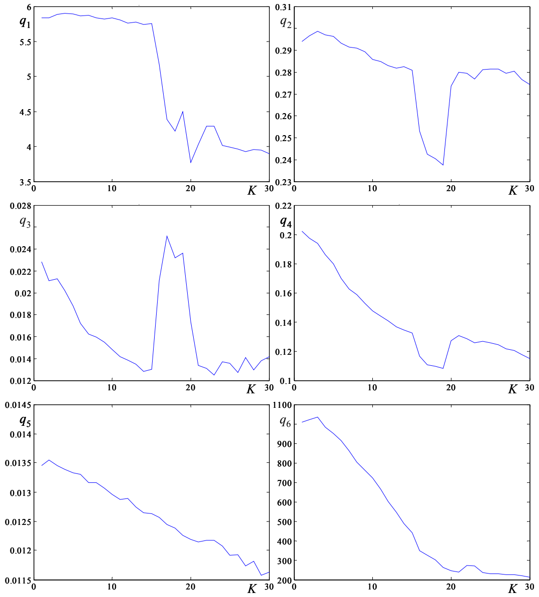


Figure 3: The quality measure values, *qi*, *i* = 1…6, in iterative optimization steps, *K* = 1…30.

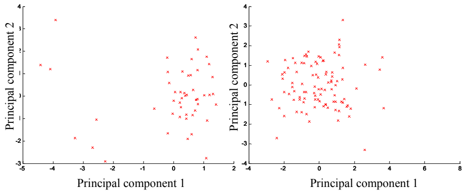


Figure 4: Distributions of *q* values projected to the plane spanned by the two major principal components, *K* = 18 (left) and *K* = 30 (right).

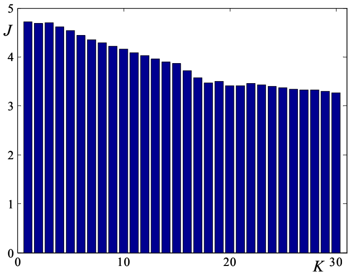


Figure 5: The values of the cost function, *J*, in the iterative optimization, *K* = 30.

The IRT method succeeded in improving the process performance regarding the six quality measures (see Figure 3). E.g., the kappa number deviation from the setpoint, q_1, diminished from about 6 to 4 units, and the absolute value of the H factor prediction error, q_6, was reduced from about 1000 to 200 units. For softwood pulp the kappa number is typically tried to keep within ±2-3 units range from the target. In that sense the control is not yet satisfactory but the tuning succeeded to improve its performance notably. The values of *q* include some stochastic variation and therefore the trends are not monotonically descending. Also occasional “outliers” can be perceived in the values of the quality measures in the global iteration steps 15-20. The convenient linear model was justified with the assumption of Gaussian data. The distributions of the quality measure data in global steps *K* = 18 and K = 30 are shown in Figure 4. Obviously the distribution in the left figure does not fulfil the Gaussianity assumption, which explains the outliers in Figure 3.

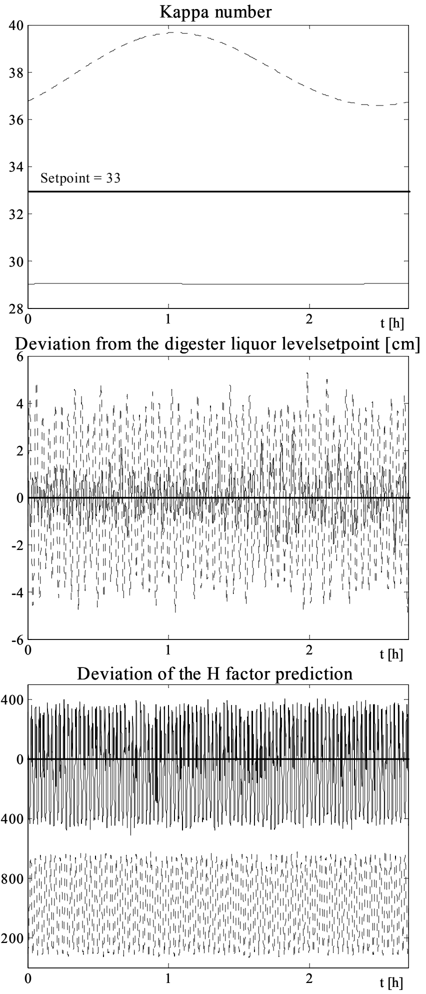


Figure 6: Simulation results with initial and tuned parameters, *K* = 1 (dotted) and *K* = 30 (solid).

The success of the optimization can be followed also from the values of J (see Figure 5). It can be seen how the conflicting targets finally started to slow down the tuning procedure. If the objectives are not met at the end, one has to reconsider the weighting of q_i. It is the only way to continue the optimization among Pareto optimal solutions. Alternatively, reformulation of the quality measures may yield in better results. Examples of the improved performance are presented in Figure 6. The fluctuation of the kappa number has stopped, variance of the digester liquor level has diminished and the H factor prediction has improved tremendously.

4 Conclusions

The IRT method offers a holistic approach to manage large complex systems, like control systems of industrial processes. Dimensional complexity in systems becomes manageable by using statistical multivariate methods. The IRT method opens up new possibilities for control engineering: Different control structures can be tuned to their maximum performance and compared with each others before implementation.

The stochastic nature of the optimization problem advocates the use of methods that diminish the effects of noise. In that sense IRT as a local optimization method using linear regression to approximate the cost function locally seems to be a more tempting approach than, e.g., genetic algorithms.

The development work on IRT method continues. One object of concern is the minimization of the cost function evaluations during the tuning procedure, since one function evaluation of (6) requires reasonably long time series data from the system. This increases the time and computational power needed for the tuning procedure. One way to minimize the amount of data is to involve a pseudo-random parameter space sampling in the data acquisition instead of completely random variation.

Future research will also include studies on simulation model tuning. The objective is to tune the model parameters to give the best output prediction possible with the chosen model structure. An example of the model parameter tuning is given by Halmevaara and Hyötyniemi (2006).

The goals of evolution in technical systems differ from those of natural neocybernetic systems. The emergy transfer maximization does not result in performance that satisfies the ever-tightening standards of profitability in process industry. Instead, maximization of the environment with the intended system functionalities gives practically applicable working practices for control engineering. The tuning of environmental variables towards better system performance along the axes of freedoms (determined by the PLS latent basis) in the parameter space can be seen as a directed technical evolution of the system.

References

Gelman, A., J.B. Carlin, H.S. Stern and D.B. Rubin (2004). *Bayesian Data Analysis*, 2nd Edition, Chapman & Hall / CRC Press, Boca Raton, FL, USA.

Gill, P.E., W. Murray and M.H. Wright (1981). *Practical Optimization*, Academic Press, London, Great Britain.

Gustafson, R.R., C.A. Sleicher, W.T. McKean and B.A. Finlayson (1983). Theoretical Model of the Kraft Pulping Process. *Industrial & Engineering Chemistry Process Design and Development*, 22, 87-96.

Haavisto, O. and H. Hyötyniemi (2006). Neocybernetic Modeling of a Biological Cell. (To be published in) *Proceedings of the 9th Scandinavian Conference on Artificial Intelligence SCAI 2006, Espoo, Finland.*

Hägglund, T. (1999). Automatic detection of sluggish control loops. *Control Engineering Practice*, 7, 1505-1511.

Halmevaara, K. and H. Hyötyniemi (2004). Iterative Simulation Based Multivariate Control Parameter Tuning Method. *Proceedings of the 5th EUROSIM Congress on Modeling and Simulation*, Paris, France.

Halmevaara, K. and H. Hyötyniemi (2005). Performance Optimization of Large Control Systems – Case Study on a Continuous Pulp Digester. *Proceedings of the 15th IFAC World Congress*, Prague, Czech Republic.

Halmevaara, K. and H. Hyötyniemi (2006). Data-based parameter optimization of dynamic simulation models. (To be published in) *Proceedings of the 47th Scandinavian Simulation Conference SIMS'06, Helsinki, Finland.*

Harris, T.J., C.T. Seppala and L.D. Desborough (1999). A review of performance monitoring and assessment techniques for univariate and multivariate control systems. *Journal of Process Control*, 9, 1-17.

Hyötyniemi, H. (2001). *Multivariate Regression – Techniques and Tools*, Helsinki University of Technology, Control Engineering Laboratory Report, Helsinki, Finland.

Hyötyniemi, H. (2002). On Emergent Models and Optimization of Parameters. *Proceedings of 42nd Scandinavian Simulation Conference SIMS'02*, 45-50. Oulu, Finland.

Hyötyniemi, H. (2006). *Neocybernetics in Biological Systems*. (To be published in) Helsinki University of Technology, Control Engineering Laboratory Report, Helsinki, Finland.

Juslin, K. and M. Paljakka (1997). APROS - A Multifunctional Modelling Environment, *Proceedings of the 2nd CSNI Specialist Meeting on Simulators and Plant Analysers*, Espoo, Finland.

Pindyck, R.S. and D.L. Rubinfeld (1991). *Econometric Models and Economic Forecasts*, 3rd edition, McGraw-Hill, New York, USA.

Royston, J.P. (1982). An extension of Shapiro and Wilk's W test for normality to large samples. *Applied Statistics*, 31, pp. 115-124.

Thode, H.C. (2002). *Testing for Normality*, Marcel Dekker, New York, USA.

Thornhill, N.F. and T. Hägglund (1997). Detection and diagnosis of oscillation in control loops. *Control Engineering Practice*, 5, 1343-1354.

Emergent Behavior in Sensor/Actor Networks in the Neocybernetic Framework of Elastic Systems

Michael Sailer
[⋆]Helsinki University of Technology
Control Engineering Laboratory
P.O. Box 5500, FIN-02015 TKK, Finland
sailer.mike@web.de

Heikki Hyötyniemi
[†]Helsinki University of Technology
Control Engineering Laboratory
P.O. Box 5500, FIN-02015 TKK, Finland
heikki.hyotyniemi@hut.fi

Abstract

If one wants to describe emergent behavior in real-life systems, the framework of *neocybernetics* offers good possibilities. Neocybernetics deals with the modeling and control of complex real-life systems and it can be shown that emergence appears in terms of self-stabilization and self-organization. In order to gain reasonable results, multivariate statistical tools are applied. A new approach in the fields of neocybernetics are *elastic systems*. These systems are defined by a strong mutual interconnection with their environment in the form of material and/or information flow from and into the system. Elastic systems are better described by dynamic than static balances; if external forces influence the system, these forces are compensated by finding another equilibrium state, but if the forces are taken away again, the original system state is restored. The new approach offers the opportunity, for example, to control systems with distributed sensor/actor networks. It can be stated that emergent behavior pops up, even if there are no explicit communication structures among the sensor/actor nodes. This paper introduces the ideas of elastic systems in terms of simulations of the distributed control of an elastic steel plate.

1 Introduction

In the field of Artificial Intelligence, the need for decentralized approaches is increasing — it has even been said that the letters AI today mean Ambient Intelligence, or Agent Intelligence. Even the basic books in AI have adopted such agent perspectives (Russell and Norvig, 2002). However, despite the immediate need for conceptual tools for managing decentralized architectures, there exists no general theory: The applications are pragmatic software products that cannot be analyzed in mathematically efficient ways. The problem is that of differing semantics in different fields — the ontologies and the goals are domain-specific.

What if one could define some universal measure for semantics that would be applicable in any field? Indeed, this is pursued in *neocybernetics*, where the key concept is *information* as interpreted in a very pragmatic way: Information is manifested in (co)variation among data items (Hyötyniemi, 2006b). The goals of the actors are assumed to be always equal — they want to survive, selfishly trying to capture as much of the available information as possible. It turns out that *self-regulation* emerges because there is feedback through the environment: Resources being exploited are exhausted. What is more, there is *self-organization* in the system in terms of multivariate statistical constructs (principal subspace analysis, PSA), principal components emerging from the adaptation processes, making it possible to analyze the system also on the holistic level (Hyötyniemi, 2006b).

It is a nice coincidence that the goals of the individual actors — exploitation of information in the form of variation — happens to equal the goals of regulating controllers. This makes it possible to implement very simple agent-based controllers: There is no need for explicit communication among agents.

This approach of emphasizing the connection to one's environment resembles the Brooksian subsumption thinking (Brooks, 1991): The world itself is the best model of the world, and observation data is directly employed for appropriately adaptating the model structures. Even though such thinking cannot be applied in too complicated tasks, in special cases, like in control tasks where the simplified view of semantics is appropriate, it can be beneficial. There is

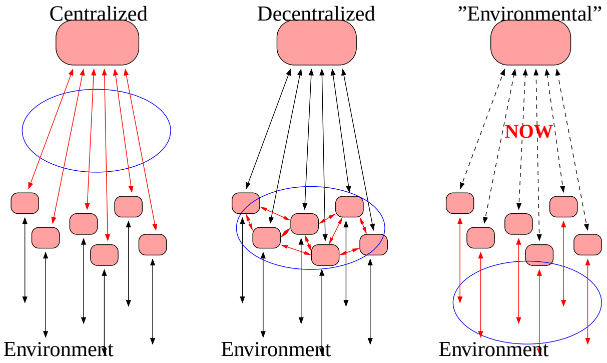


Figure 1: Communication and coordination among low level nodes.

no explicit communication among agents needed — whereas in traditional agent systems the operation is either centrally coordinated or negotiated among the agents, now it is the agents that only look after themselves (Figure 1).

A new interpretation of neocybernetics describes complex systems as *elastic systems* (Hyötyniemi, 2006a). These systems are in strong interconnection with their environment and characterized by material flow from and into the system; the system can not be regarded isolated, as it is embedded in its environment: changes in the environment affect the system and changes in the system itself affect the considered environment respectively. In the framework of elastic systems, decentralized control of elastic deformable systems — following the idea of dynamic equilibria — can be realized with distributed sensor/actor networks. As an example a steel plate, exposed to different external forces, is studied in a simulation environment (Sailer, 2006). It turns out that the adaptation process stiffens the considered system. Moreover, the completely adapted system can be described in terms of *constant elasticity*; the structure is adapting towards the maximum experienced stiffness.

2 Towards intelligent behavior in sensor/actor networks

There exist many industrial processes where systems can be described in terms of partial differential equations with distributed parameters. Generally, the complete information in these systems is not measurable, as the process state is infinite-dimensional. However, a reasonable setup to capture the necessary information is that of distributed sensor/actor networks: the infinite-dimensional system state is mapped on the finite amount of applied sensor/actor nodes, maintaining the distributed characteristics of the system.

Today's challenge is the orchestration of these networks, so that emergent behavior can evolve in these systems, or, in other words, so that the network behaves in an *intelligent* way. Conventional approaches concentrate on the search of the global structure of the sensor/actor network — but if one knows this picture, the coordination can also be performed in a centralized manner. What is more, the centralized scheme has yet another disadvantage: centralized implementation often represents the originally distributed structure of the systems very poorly.

The common framework for distributed systems is the *agent* perspective Lesser et al. (2003) but these studies have so far no consistent mathematical framework. Theories are often qualitative and do not offer concrete design methods.

Neocybernetics offers a consistent mathematical framework that could be applied in truly distributed sensor/actor networks. In these networks, the single sensor/actor nodes are capable of some limited computation and communication structures could be implemented for example with simple protocols. In these networks, the human can be detached from the process, the network is acting in a completely autonomous manner, assuming that the simple view of information can be accepted (Hyötyniemi, 2006b). One application example is the research of emergent coordination of distributed sensor networks (Hyötyniemi, 2004).

In the case of elastic systems, the balancing of the system is implemented implicitly, through feedback from the environment. In such environments, emergence can be reached with no explicit communication among the sensor/actor nodes whatsoever; every node is working on its own, but still, intelligent behavior can be found as shown in 5. In what follows the basic components of the networks, the agents, are called nodes. The nodes are identical and they can measure their environment (sensors) and influence the environment respectively (actors).

3 Elastic systems

This chapter introduces the ideas of elastic systems in general and its mathematical framework is presented.

3.1 Feedback through environment

In traditional control, the information of the participating nodes is collected and evaluated by central units in order to coordinate the system in the desired direction. Distributed modeling approaches concentrate on explicit communication structures among distributed actors. It turns out that new thoughts on elastic systems lead to emergent behavior also in truly distributed systems. The essential difference compared to the other two approaches is the existence of an implicit feedback structure directly through the environment itself. Figure 1 presents the new framework of elastic systems compared to the traditional approaches. The steel plate gives an excellent example for the understanding of this implicit feedback. If there are forces acting on the plate, these forces induce counter forces, according to the measured deformations of the applied sensor/actor nodes. These counteracting forces are directly measurable by other sensor/actor nodes, as they change the steel plate deformation, although the nodes do not have any active communication structure.

3.2 Mathematical considerations

Elastic systems are systems in dynamic equilibrium; if some *forces* are applied on the system, these forces cause *deformations* in the system, but if the forces vanish, the original state of the system is restored again. In the context of elastic systems the physical interpretation of forces and deformations can be different but the overall behavior of the system remains the same: For example, in chemistry this phenomenon is called *Le Chatelier Principle*. There, the dynamic equilibrium of chemical reactions drifts in another balanced state in order to counteract external changes. This abstraction level of elasticity offers a wide scope of applications (Hyötyniemi, 2006a).

The dynamic equilibrium of an elastic system — or the visible, emergent system model — is often expressed as

$$A\bar{x} = Bu. \tag{1}$$

Here, $\bar{x} \in \mathcal{R}^n$ describes the balanced system states x ($\lim_{t\to\infty} x = \bar{x}$), dependent on the measurable system input $u \in \mathcal{R}^m$, with $n \leq m$. In dynamic equilibrium, the system vector $\bar{x}$ can be solved explicitly, assuming that A^{-1} exists and there holds

$$\bar{x} = A^{-1}Bu = \phi^T u. \tag{2}$$

As elastic systems are in strong interconnection with their environment, caused by a feedback $-\varphi x$ from the system back into the environment, there exists a balanced state $\bar{u}$ for the system environment u and there holds

$$\bar{u} = \underbrace{u}_{\text{actual environment}} - \underbrace{\varphi\bar{x}}_{\text{feedback}} . \tag{3}$$

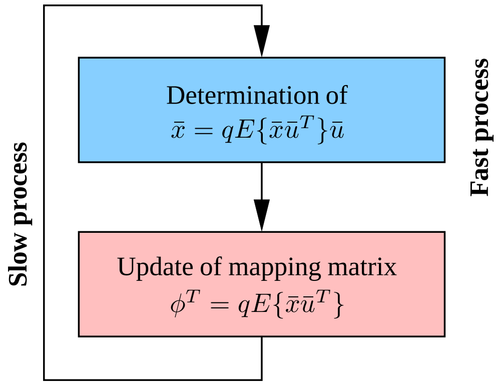

Determination of
$\bar{x} = qE\{\bar{x}\bar{u}^T\}\bar{u}$
Update of mapping matrix
$\phi^T = qE\{\bar{x}\bar{u}^T\}$
Slow process
Fast process

Figure 2: Flowchart of the adaptation process.

As the system sees only the balanced environment $\bar{u}$, (2) results in

$$\bar{x} = \phi^T\bar{u}. \tag{4}$$

The mapping matrix ϕ^T can be motivated by the *evolutionary fitness* (Hyötyniemi, 2006b) of complex real-life systems and then there follows:

$$\phi^T = qE\{\bar{x}\bar{u}^T\}. \tag{5}$$

Matrix $E\{\cdot\}$ denotes the covariance matrix. The constant factor q describes the strength of the interconnection between system and environment and can be interpreted as a *stiffness ratio*.

It is the mapping matrix ϕ^T that determines the emergent behavior. If one studies (4) and (5) closer, it can be seen that the solution for the mapping matrix has to be found iteratively as the balanced system states $\bar{x}$ and the balanced environmental states $\bar{u}$, determining the covariance matrix $E\{\bar{x}\bar{u}^T\}$ are not known beforehand. Therefore, the adaptation can be described as follows (see Fig. 2):

1. A fast adaptation finds the balanced system state (4), when external forces are applied on the system, causing deformations.

2. A slower process determines the mapping matrix ϕ^T. The system learns from different applied external forces and the overall balance is found, after the mapping matrix ϕ^T has adapated and does not change any longer — a *second order balance* is found.

For the validity of the adaptation process it is assumed that the change of the external forces is much slower than the determination of a balanced system state with (4). How the streamlining is realized can be read in (Sailer, 2006) and (Sailer and Hyötyniemi, 2006).

With the use of (5), the adapted system results finally in a balance, where the covariance matrix $E\{\bar{u}\bar{u}^T\}$ has the property

$$\frac{1}{q} I_n = \bar{\theta}^T E\{\bar{u}\bar{u}^T\}\bar{\theta}, \tag{6}$$

and there holds $\bar{\theta}^T\bar{\theta} = I_n$. It turns out that the columns in θ span the subspace determined by the n most significant principal components of $E\{\bar{u}\bar{u}^T\}$ (Hyötyniemi, 2006b). For more information on principal components, see (Basilevsky, 1994). Besides this equalization of variances, it can be seen that all eigenvalues $\bar{\lambda}_j, j = 1, ..., n$ of the closed loop equal $1/q$, meaning that the system becomes stiffer compensating the external applied forces. In this context one can speak of *constant elasticity* of the system.

In (Hyötyniemi, 2006b) it is shown that the latent system variables $\bar{x}$ can be chosen freely as $\bar{x}'$. This property makes it possible to apply *active control signals* as system variables no matter how the controls affect the environment. If there were complete information, all sensor/actors knowing all inputs, they still together implement principal subspace analysis. As this happens regardless of the mechanisms of actuation, it turns out that a cybernetic set of controllers *changes the environment* to match the theoretical model.

In this paper this sensor/actor scheme is simulated to set up the distributed control of an elastic deformable steel plate. Such a mechanical environment offers a nice test bench as the effects are delayless, and the local balances after force/counterforce application are found immediately (in principle — when the environment is simulated, there are additional issues, like balance problems). Here the information is not complete; it is assumed that each actor only is aware of its own local sensor. This means that the behaviors become strongly localized, but here, too, inter-agent organization takes place as there is blurring between the effects of individual actors along the steel plate.

4 Distributed control of an elastic deformable steel plate

A control scheme for a steel plate using a sensor/actor network, consisting of m sensor/actor nodes, was implemented. The measured steel plate deformations s define the environmental variables u, caused by some external applied forces F_{ext}. The induced actor forces F_{act} are the latent freely chosen system variables x' so that there holds:

$$\left\{ \begin{array}{rcl} F_{act} & = & x' \\ s & = & u \end{array} \right. \tag{7}$$

If one assumes that there are no communication structures among the participating sensor/actor nodes, the mapping matrix ϕ^T is realized diagonal. In this case, the single actor forces $\bar{F}_{act,i}$ are only determined by their own measured sensor information $\bar{s}_i$.

Despite this simple learning, a global emergence can be observed in the simulations: with the introduced local adaptation and the fact that there are m sensor/actor nodes applied on the steel plate, there follows from (6)

$$\frac{1}{q} I_m = E\{\bar{s}\bar{s}^T\}, \tag{8}$$

with $E\{\bar{s}\bar{s}^T\}$ diagonal. With this relationship, the coupling coefficient q can be used in order to predict the emergent steel plate deformation after the system has fully adapted. The variances $E\{\bar{s}_i^2\}, i = 1, ..., m$ are directly determined by q. Therefore two different behaviors can be stated out in the adapted system:

- There is a loss in the variations $E\{\bar{s}_i^2\}, i = 1, ..., m$, meaning that the system is getting stiffer.
- The variances are all equalized by the factor $1/q$.

This observation is crucial, as the coupling coefficient q determines already *before* the start of the adaptation process, how the emergent shape of the (observed) steel plate looks like.

5 Simulation

This chapter shows in simulation examples, how emergent behavior evolves in case of the adaptation process of the steel plate, and interpretations are offered afterwards.

5.1 Results

The studied square-form steel plate has the dimensions $0.8 \times 0.8[m]$, and it is $0.01[m]$ thick. On the plate four sensor/actor nodes are placed as shown in Figure 3. The applied external forces are uniformly distributed between $-550[N]$ to $-600[N]$ for the second order adaptation. For the fast process these forces are constant (denoted as set l in the figures). In total 10000 different sets of external forces are used for the system adaptation. The studied steel plate is fixed on opposite sides. At these sides, the displacement of the steel plate is zero ($\Delta s = 0$).

Figure 4(a) shows the development for the measured displacements of the sensors during the ongoing adaptation process and Figure 4(b) the developing actor forces respectively. It can be seen that the system is getting stiffer with the ongoing adaptation process, as the actor forces are growing and the measured sensor displacements are getting smaller. The property of constant elasticity, mentioned earlier can be seen; the variances $E\{\bar{s}_i^2\}$ are equalized with the factor $1/q$. In the example, the coupling coefficient q has a value of $q = 4.5 \cdot 10^7 [1/m^2]$. According to (8), the variances $E\{\bar{s}_i^2\}$ are

$$E\{\bar{s}_i^2\} = 1/q = 2.22 \cdot 10^{-8} [m^2], \quad i = 1, ..., m. \tag{9}$$

As the variation level in the external forces is low, the expectation values of the measured displacements $E\{\bar{s}_i\}$ can be approximately determined directly from (9) as

$$E\{\bar{s}_i\} = 1/\sqrt{q} \approx \pm 1.49 \cdot 10^{-4} [m], \quad i = 1, ..., m. \tag{10}$$

Figure 4(a) shows that the expectation values $E\{\bar{s}_i\}$ are equalized on the level determined by (10); before the adaptation process takes place the sensor displacements are differing for the single nodes. Before convergence, there is exponential growth in the parameters and in the sensor forces; only after the effects in the system are noteworthy, there is feedback from the environment and the adaptation process stabilizes.

Figure 4(b) reveals different developments in the actor forces; those sensor/actor nodes where higher or more frequent deformations are measured are evolving stronger than nodes that are closer to the fixed boundaries or that are less deformed. Actor forces 3 and 4 are developing stronger than the forces 1 and 2 as they are closer to the applied external forces (see Figure 3). Force 1 is completely fading away, meaning that the sensor/actor node is not actively participating in the compensation after the system is fully adapted. In this case, sensor/actor node 1 can be described as *sub-cybernetic*; the global emergent goal of constant elasticity is reached, but the node does not play a particular role in the adapted system.

If the coupling coefficient q increases, all nodes are becoming active in the adapted sensor/actor network. Figures 5(a) and 5(b) show the results in the case of $q = 7.0 \cdot 10^7 [1/m^2]$. Here, once again, the final goal as determined by the coupling coefficient q is reached as $E\{\bar{s}_i\} = 1/\sqrt{q}$ holds, but all nodes are participating. In this case one can speak of a *fully cybernetic* system. A deeper analysis on the coupling coefficient q and the influence on the emergent pattern in the sensor/actor networks can be read in Sailer (2006).

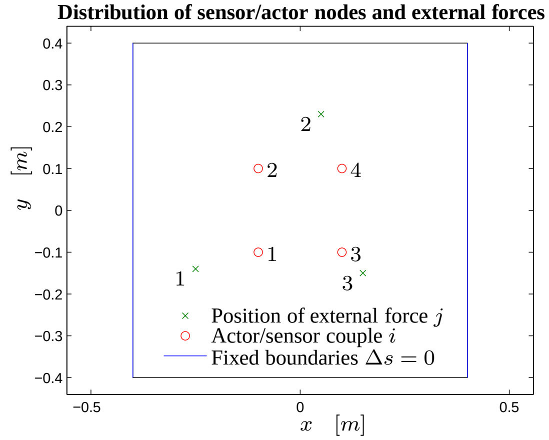


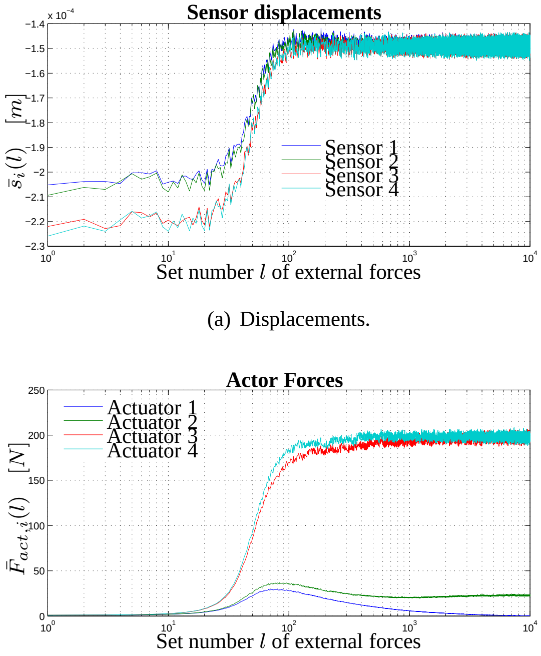
Figure 3: Steel plate setup.

5.2 Interpretations

It has to be emphasized that despite the simple learning structure of the sensor/actor network emergent behavior can be observed in the elastic deformable steel plate. Although there are no communication structures among the sensor/actor nodes, the system finds a state of constant elasticity after the adaptation process. The sensor/actor nodes are communicating implicitly, as induced actor forces are measured as deformation changes by the other sensor/actor nodes.

If one compares the adaptation with conventional control tasks, the advantage of the proposed approach is obvious: there is no need of long lasting system identifications to achieve emergent behavior with the applied sensor/actor network. The nodes are adapting without the explicit knowledge of the global picture.

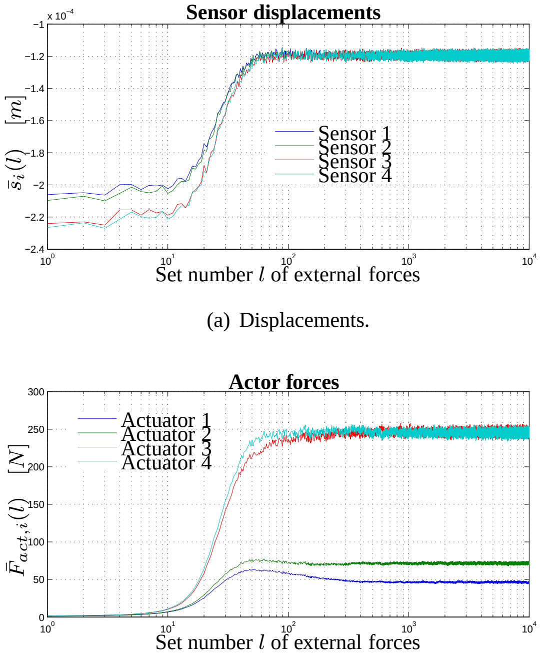
As one takes a closer look at the development of the actor forces during the ongoing adaptation process, one can observe that the behavior of the actors can be described as somehow *intelligent*. The global result of the adaptation is the stiffening of the steel plate on a constant level. As it can be seen in the first simulation example, nodes can vanish completely. These nodes are not necessary for the global emergence and therefore are dying. Their faith depends on how close they are to the *source of concern*. Sensor/actor nodes at more flexible positions, or at positions with more frequent or stronger occurring forces are developing stronger than nodes that are closer to boundaries or where external forces cause smaller deformations.



(a) Displacements.

(b) Actor Forces.

Figure 4: Adaptation process of the steel plate, $q = 4.5 \cdot 10^7 [1/m^2]$.



(a) Displacements.

(b) Actor Forces.

Figure 5: Adaptation process of the steel plate, $q = 7.0 \cdot 10^7 [1/m^2]$.

It is easy to make hypotheses here. In biological systems such stiffening behavior can be observed as well: human cells are growing stronger or more cells are produced, if more frequent or stronger external signals are influencing specific cell areas. Examples are the human skin or muscle cells. In biology, these phenomena are called hypertrophy and hyperplasia. The motivation of the mapping matrix ϕ^T, as defined in (5), describing local evolutionary learning, seems to achieve reasonable results that can be interpreted in a biological context respectively.

Also in less concrete cybernetic systems, such stiffening is commonplace: the controls become stronger, and degrees of freedom vanish. It does not matter whether such "evolution" is implemented by nature, or by human, as in social systems.

When studying *supply* and *demand* in product markets, for example, the steel plate analogies are again applicable. In economy, the real world of customer needs (the "steel plate") is infinite-dimensional and poorly structured as seen from outside. When an actor/sensor couple is applied in some location there, however, or when a product with specific properties is introduced, the market deforms to balance the supply and demand. Products that are poorly located — if there is no demand (deformation) or if there are other better products nearby — soon fade away. As studied closer in (Hyötyniemi, 2006b), such analogies also make it possible to study the age-old philosophical questions in a new setting:

> "Is there any demand if there is nobody to supply?"

References

A. Basilevsky. *Statistical Factor Analysis and Related Methods*. John Wiley & Sons, New York, 1994.

R.A. Brooks. How to build complete creatures rather than isolated cognitive simulators. In K. VanLehn, editor, *Architectures for Intelligence*, pages 225–239. Lawrence Erlbaum Assosiates, Hillsdale, NJ, 1991.

H. Hyötyniemi. *Emergent Coordination in Distributed Sensor Networks*. Finnish Artificial Intelligence Conference SteP'04, 2004.

H. Hyötyniemi. *Elastic Systems — Another View at*

Complexity. Scandinavian Conference on Artificial Intelligence (SCAI), 2006a.

H. Hyötyniemi. *Neocybernetics in Biological Systems*. Helsinki University of Technology, Control Engineering Laboratory, 2006b.

V. Lesser, C.L. Ortiz Jr., and M. Tambe. *Distributed Sensor Networks — A Multiagent Perspective*. Kluwer Academic Publishers, Boston, 2003.

S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach (2nd edition)*. Prentice Hall, 2002.

M. Sailer. *Distributed Control of a Deformable System - Analysis of a Neocybernetic Framework*. *Diploma Thesis*, Helsinki University of Technology, Control Engineering Laboratory, 2006.

M. Sailer and H. Hyötyniemi. *Analysis and Simulation of the Distributed Control of Elastic Systems in the Neocybernetic Framework*. Conference on Simulation and Modeling (SIMS), September, 2006.

Elastic Systems: Case of Hebbian Neurons

Heikki Hyötyniemi[⋆]

[⋆]Helsinki University of Technology
Control Engineering Laboratory
P.O. Box 5500, FIN-02015 TKK, Finland

Abstract

As the neuronal (and cognitive) processes are, after all, so well known, they offer a nice testbench for complexity theories, and they are a nice prototype for understanding behaviors in cybernetic systems in general. It turns out that when applying the framework of *Hebbian neurons,* many observed brain functionalities can be attacked, including *sparse coding* on the low level, and *chunking* on the higher one. Even the mysteries of *causality* and *mind vs. matter* can perhaps be given new perspectives.

1 Introduction

Modeling of neurons is a nice application area — on the other hand, it is well-known (at least much-studied), but on the other, the neuronal system seems to be among the most complex ones to be understood. Brains and mental functions, or matter and mind, seem to be incompatible, and are a subject of age-old disputes. There are different levels, and emergence of higher-level structures from lower-level ones is evidently necessary — but, what is more, the different levels seem to be related to completely different *phenospheres.* First, there is the ecosystem of neurons; second, there is the infosystem of signals; and last, there is the "ideasystem" of concepts. All of these reside in the same physical medium, and powerful conceptual tools are needed to detect and distinguish between the phenomena.

The brain and cognition have long been seen as cybernetic systems. However, there are disputes: Heinz von Foerster coined the term *second-order cybernetics* and claimed that a cybernetic system (observer) cannot study another cybernetic system of the same complexity.

Here, it is claimed that *neocybernetics,* and, specifically, analysis of *elastic systems* offers a consistent framework that can capture brain and mind related phenomena in all of the phenospheres and in all of the levels. In the neocybernetic framework, all relevant constructs are *statistically determined dynamic equilibria,* being *attractors of the dynamic processes in the data space.* Hierarchies of higher-order cybernetics are reduced back to analysis of data properties.

From the point of view of AI, the key point in elastic systems is that when applying such a strong framework, it is possible to combine quantitative data processing and still reach qualitatively relevant models. This claim will be elaborated on in this paper, being continuation to the presentation in (Hyötyniemi, 2006a).

2 Modeling of neurons

It turns out that the neuronal system is *isomorphic* with the general elastic system. The main properties of such systems remain intact, but there are some issues that need to be studied from another point of view.

2.1 Hebbian perceptrons

Neurons are extremely complex electrochemical entities, and interactions among them are based on asynchronous, pulse-coded signals. In practice, such neuronal systems cannot be studied in detail, but simplifications are necessary. It turns out that when one abstracts away the time axis, studying average activity levels instead of the individual pulses, one can reach practical models — this is the approach in practically all artificial neural network structures (Haykin, 1999). In its basic form, the activities of the environment $\bar{u}_j$, where $1 \leq j \leq m$, and the activities of the neuron grid $\bar{x}_i$, where $1 \leq i \leq n$, can be coupled applying the *perceptron model,* so that $\bar{x}_i = f\left(\sum_{j=1}^{m} w_{ij}\bar{u}_j\right)$. Here, the activity of a neuron is simply a weighted sum (weighting factors w_{ij} being the synaptic strenghts) of the input activities, being further modified by the activation function f. When such expressions for the whole neuron grid are collected together, and if the nonlinearity is ignored, one has the simple matrix formula

$$\bar{x} = W\bar{u}, \tag{1}$$

where W is the matrix of synaptic weights mapping from the input vector $\bar{u}$ onto the neuronal state vector $\bar{x}$. This simplistic formulation is employed here, because it is all one needs. It directly complies with the elastic systems framework, with $\bar{\phi}^T = W$, and it fulfills the neocybernetic ideals[1]. The model (1) defines a static mapping between sets of variables, so that variables are coupled in terms of a set of constraints. Again, this pattern can be extended applying elasticity assumptions and dynamic tension considerations into another forms. The balance pursuit property of the neuronal system is crucial, giving rise to elastic master-slave behavior: No matter what is the input $\bar{u}$, neuronal state $\bar{x}$ finds its equilibrium. The properties of the converged system are reduced back to statistical properties of the environment.

To keep the system balanced, non-idealities need to be employed. The normal approach is to introduce the non-ideality in the form of nonlinearity, but following the neocybernetic guidelines, another kind of non-ideality is used: Here it is assumed that the input u is exhausted as it is exploited, leaving only the residue $\bar{u}$ visible. The assumption of information flows always being carried on a material flow is a general principle, and it constitutes now the stabilizing (linear) feedback between the system and the environment.

In the neuronal system the assumed evolutionary dynamics is also nicely manifested. According to the Hebbian learning principle (Hebb, 1949) it has been observed that for real neurons there holds

> If the the input activity (now $\bar{u}_j$) and neuronal activity (now $\bar{x}_i$) correlate, the synaptic weigth becomes stronger.

This exactly equals the evolutionary goal of an elastic system of maximizing the coupling with the environment by adapting w_{ij} in the direction of $\mathrm{E}\{\bar{x}_i\bar{u}_j\}$, as studied in (Hyötyniemi, 2006a).

2.2 Sparse coding

What happens when Hebbian learning boosted with feedback takes place in the neuron grid?

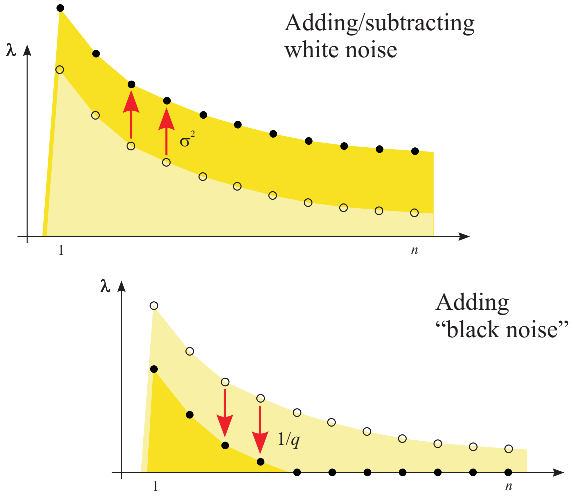


Figure 1: "Black noise" as compared to white noise: Variance structure of data in different directions

In (Hyötyniemi, 2006a) connections among $\bar{x}$ and $\bar{u}$, and among $\bar{x}$ and Δu were studied. When studying the theoretical mapping between $\bar{x}$ and the original undisturbed input u, it turns out (see Hyötyniemi (2006b)) that the eigenvalues of $\mathrm{E}\{\bar{x}\bar{x}^T\}$ can be expressed in terms of the n most significant eigenvalues λ_j of $\mathrm{E}\{uu^T\}$. Specially, if the coupling coefficients q_i and b_i are different for different neurons, the i'th eigenvalue (or latent variable variance) becomes

$$\frac{\sqrt{q_i\lambda_j} - 1}{b_i}, \tag{2}$$

indices i and j being ordered randomly. This reveals that there must hold $q_i\lambda_j > 1$ for that input variance direction to remain manifested in the system activity — if this does not hold, variable $\bar{x}_i$ fades away. On the other hand, for the modes fulfilling the constraint, interesting modification of the variance structure takes place; this can best be explained by studying a special case. Assume that one has selected $q_i = \lambda_j$ and $b_i = 1$ for all pairs of i and j. Then the corresponding variances become $\lambda_j - 1$ (see Fig. 1). In each direction in the data space, the effect of the system is to bring the variance level down if it is possible. Analogically, because white noise increases variation equally in all directions, one could in this opposite case speak of "black noise".

What are the effects of this addition of black noise in the signals? First, it is the principal subspace of u that is spanned by the vectors $\bar{\phi}_i$. But assuming that this subspace is n dimensional, there exist many ways how the basis vectors can be selected, and some of the selections can be physically better motivated.

[1] Note that nonlinearity is traditionally included in the perceptron model for pragmatic reasons: First, nonlinearity promises more functionality, and nonlinearity assures stability (compare to *Oja's rule*). In the neocybernetic spirit, linearity is pursued, because only then scalability of the models can be reached. It is the dynamic feedback structures that provide with nontrivial functionalities — like *self-regulation* and *self-organization*

For example, in *factor analysis* the PCA basis vectors are *rotated* to make them aligned with the underlying features, and the same idea takes place in *independent component analysis.* In factor analysis, it is assumed that the underlying features can be characterized in mathematical terms applying the idea of *sparseness:* When a data vector is decomposed, some of the latent variables have high scores while the others have low scores, increasing the differences among latent variable variances. This goal can be iteratively implemented in terms of criteria like *varimax* or *quartimax,* etc. In its extreme form, sparsity means that there are only a few of the candidates employed at a time, and the goal of modeling, rather than being minimization of the number of overall model size, it is the minimization of *simultaneously active constructs.* This means that the total dimension of the latent basis n can even become higher than the dimension m of the input data, the basis being *overcomplete.*

As shown in Figure 2, the *Hebbian feedback learning offers an efficient approach to achieving sparsity-oriented basis representation of the PCA subspace.* Whereas the overall captured variation (shown both in lighter and darker color in the figure) is not changed by orthogonal rotations, the variation over the bias level (shown in darker color) *can* be changed. As the nominal PCA approach typically distributes variation more or less evenly along each latent variable, it is most of the variation that remains below the threshold level; now, as it is the area above the threshold level that is maximized, non-trivial basis representations are reached.

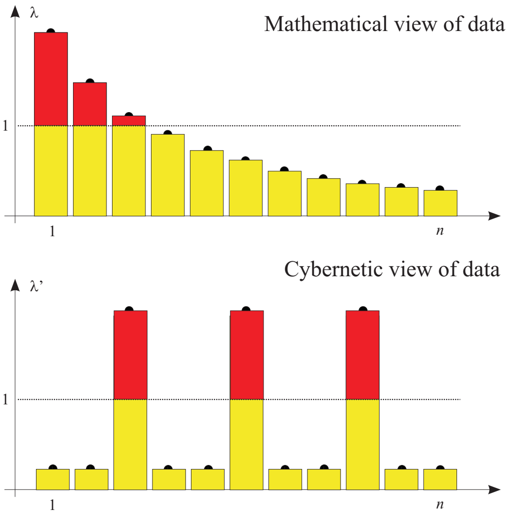


Figure 2: Schematic illustration of how black noise results in sparsity pursuit: Area above the threshold is maximized

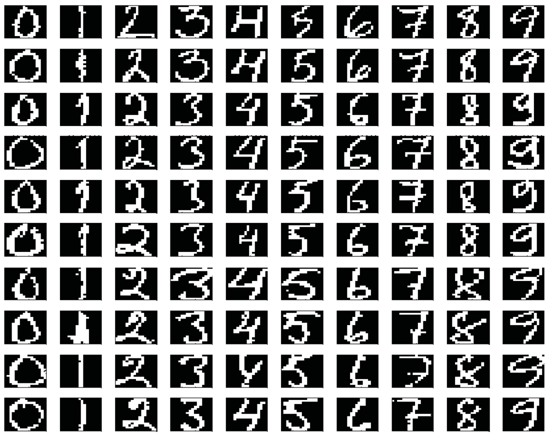


Figure 3: Examples of handwritten digits

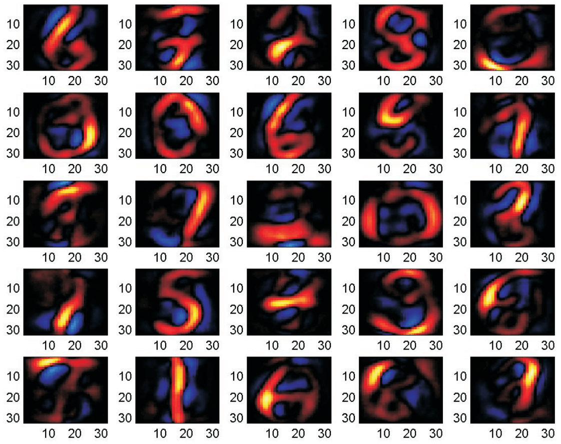


Figure 4: The 25 sparse components extracted from the handwritten digits. The 1024-dimensional vectors $\bar{\phi}_i$ are projected back onto the two-dimensional canvas, vector entries with high positive values being presented using lighter colors following the "hot/cold" color map. Clearly, different kinds of "strokes" emerge as separate memory representations. Ordering of the sparse components is random, but different runs result in essentially the same components being extracted

Figure 4 illustrates the sparse coding behavior of the feedback Hebbian neuron grid. As data material, there were over 8000 samples of handwritten digits (see Fig. 3) written in a grid of 32 × 32 intensity values (Laaksonen, 1997). The 1024-dimensional intensity vectors were used as data u, and each of the latent variables $\bar{x}_i$ was kept active by appropriately controlling the coupling factors q_i. To enhance convergence, the "cut" nonlinearity was additionally employed to emphasize sparsity of coding (see Hyötyniemi (2006b)). The behaviors differed very much from principal component coding: In the beginning, something like clustering emerged, each data region being represented by a separate variable, but as adaptation proceeded, the features started becoming more orthogonal, and patterns were decomposed further. What is interesting is that this kind of "stroke coding" has been observed also in the visual V1 cortex region.

The above view of neuronal functioning can be extended to more complex data, just by interpreting the data-based constructs in new ways.

3 Towards modeling of cognition

The neocybernetic framework not only allows modeling of the coding of individual patterns, it can perhaps give tools to attack the functioning of the complete brain. There is the intuition backing up us here: The cognitive system simply *has to be* cybernetic — even in various ways (Heylighen and Joslyn, 2001).

3.1 Population of neurons

Employing the idea of looking at the neurons as a population of competing individuals, one can see the neuronal "ecosystem" as a self-regulative entity. The computer paradigm with "memory registers", etc., collapses; there is no transfer of information, no separate long-term memory or short-term memory elements, but it is all an integrated whole. No central control is necessary, nor some "operating system", it is all about distributed pursuit for resources in terms of excitation, or variation in signals. The winning neurons start representing the corresponding association structure, defining a (subconscious) "concept atom". As atomary concepts are connected to previously activated ones, sequences of concepts emerge. In the long run, when the time structure is ripped off, some kind of a *semantic net* emerges.

This all is more or less familiar — the added value, the main contribution of the neocybernetic perspective, comes from the ability of explaining how the above declarative representations change to associative ones, or how the *shift from novice to expert* can be explained. The key functionality is the self-regulation and self-organization property of the Hebbian feedback system: As the Hebbian adaptation takes place, locally and independently in each synapse, the declarative structures become swallowed in an "associative medium". As correlating concepts are appropriately connected together, following the neocybernetic adaptation principles, links of the semantic net become denser and numerically optimized, and bidirectional. All concepts are defined in terms of examples and associations to other concepts — because of the numeric, dynamic platform, the hermeneutic cycles converge to a balance of connotations.

The above process of automatisation is the key process in the mental system. But the inverse process is equally important: The high-dimensional associative representation has to be decoded into a one-dimensional representation to facilitate any information transfer, or processes like reasoning, and *thinking* in general. Such issues are not studied here — but the claim is that if the link between declarative sequential representations and associative parallel ones someday is found, ultimate homogeneity of mental functions can be reached: There is no need for special structures when mental faculties are implemented.

3.2 Role of semantics

It is assumed here that intelligence is an illusion that emerges when a large number of simple structures cumulate. The principles assumedly remain the same also on the new emergent level, so that the processes can be reduced back to processing of data. When proceeding from the level of signal processing to processing of information and knowledge, one is facing new challenges, because one needs to address issues that are the most relevant to the human mind: A cognitive model is void, its essence escapes, giving rise to *Chinese room* type arguments, if it does not somehow capture the *semantics* of the constructs. One needs to extend from the infosphere, where it was simply data (co)variation that needed to be captured, to "ideasphere".

For concrete modeling purposes, one needs to be capable of reductionistically decomposing the cascaded system hierarchy into self-contained entities. Now, assuming that these "idea atoms" emerge from lower levels, being individual cybernetic models for subdomains, how to avoid the infinite recess, concentrating on a single level, truncating the succession of models? In other words: How to assure that the data delivered to a cybernetic system constitutes a "cybernetic closure"? How to fix the grounding of semantics, or make the concrete data contain the necessary "atoms of semantics"?

The concept of semantics needs to be formalized at some level. When processing signals, information being interpreted as (co)variation, one concentrates on *contextual semantics*, where the meaning of the structures is determined in terms of their interconnections, finally reducing back to the system inputs (*natural-*

istic semantics). For a cyberenetic system, however, this kind of static definition is not enough, one once again needs to extend the studies to dynamic domain. It was dynamic balances that were the key issue in neocybernetics, and the cybernetic models are models over such equilibria. These balances need to be buried in data, or, the data needs to be made balanced.

In each state there is a tendency to move in some direction. This "flow" is proportional to the unbalanced tensions in that state, and can be utilized to quantify the counteracting forces. Such tensions are also visible in the observation data: State changes, or differences between successive states are proportional to the flow. When such derivatives are included in the data, they stand for the additional compensating forces that are needed in that state to reach balance:

$$u'(k) = \left(\frac{u(k)}{\frac{du}{dt}(k)} \right) \approx \left(\frac{u(k)}{u(k+1) - u(k)} \right). \tag{3}$$

Such "preprocessing" of observations, emphasis on changes or differences between successive ones, can also be motivated in terms of psychological and neurophysiological studies.

As an example of the relevance of the above discussion study a case where chess configurations are modeled. Chess is the "banana fly" of cognitive science, being a simple domain, but still being far from trivial. There were some 5000 configurations from real games used for modeling[2]. The coding of the configurations was carried out so that for each location on the board (altogether $8 \times 8 = 64$) it was assumed that there are 12 different pieces that can in principle be located there, and for each of them there was a separate entry in the data vectors. This means that there are altogether $64 \times 12 = 768$ binary entries in the highly redundant data vectors — and when the derivatives were included and u' was defined as in (3) the data was $2 \times 768 = 1536$ dimensional. In Fig. 5 the results are presented when 100 memory representations or feature prototypes, or chunks, were allocated for this chess coding task. These chunks $\bar{\phi}_i$, where $1 \leq i \leq 100$, were extracted from the training data, and after convergence a typical configuration was reconstructed as a weighted sum of the chunks. In the figure, the modeling results are approximatively illustrated by projecting the numeric representations back onto the discrete-valued realm.

It is interesting to note that it has been claimed that some 50000 chunks are needed to satisfactorily represent the chess board (Chase and Simon, 1973). Now the numeric nature of the chunks and inherent optimization of the representations makes it possible to reach a much more compact model for a domain.

The results are interesting, remotely reminding the mental operationing of a real chess expert: It is known that chess experts only concentrate on the "hot spots" on the board, but this kind of attention control has not been satisfactorily explained. Of course, the current experiment only studied very elementary patterns on the board, and to capture phenomena like *functional chunks*, to reach towards really "understanding" the game, one could introduce more complex (cybernetic) preprocessing of the observations:

$$u''(k) = \left(\frac{\bar{u}'(k)}{\bar{x}'(k)} \right). \tag{4}$$

3.3 Epistemology of constructs

In today's AI paradigms (like in *semantic webs* and earlier in expert systems), it seems that one is interested in *ontologies*. However, the essence of knowledge is not in the objects but it is in the ways of conceptualizing and representing them. What kind of *epistemologies* are dictated by the underlying "wetware"?

As it is assumed that it is essentially the same Hebbian perceptrons that implement all the functionalities, there is the common neural basis of the cognitive constructs, dictating their structure. The linear, sparsely combined features are best characterized as *degrees of freedom* in data space. The "conceptual spaces" are now not based on clusters in the data space but on additive axes of degrees of freedom. Because of this uniformity, it must be so that for example *categories* and their *attributes* have essentially the same kind of structure, each determining the other. The "is-a" hierarchies and "has-property" structures become unified. Also subclasses, and, specially, instances of classes, are similarly represented as interconnected degrees of freedom. This means that the framework of *fuzzy subsets* offers an appropriate epistemology for mental constructs — subclasses belong to superclasses, but also *superclasses belong to subclasses* (see Fig. 6). Normally, category properties (couplings to other categories) are stored in the prototype itself if they are not shared, but if an attribute is common to many categories, it becomes manifested as a separate structure of its own (compare to the "strokes" as atoms of relatedness in Fig. 4).

Perceptions are lower-level observations that are filtered through the mental model. This structure is not, however, purely hierarchic — higher-level perceptions also affect lower-level ones. In concrete terms, $\bar{x}_i$ determines the relevance of the concept

[2] I am grateful to Professor Pertti Saariluoma for the data material and for encouraging discussions

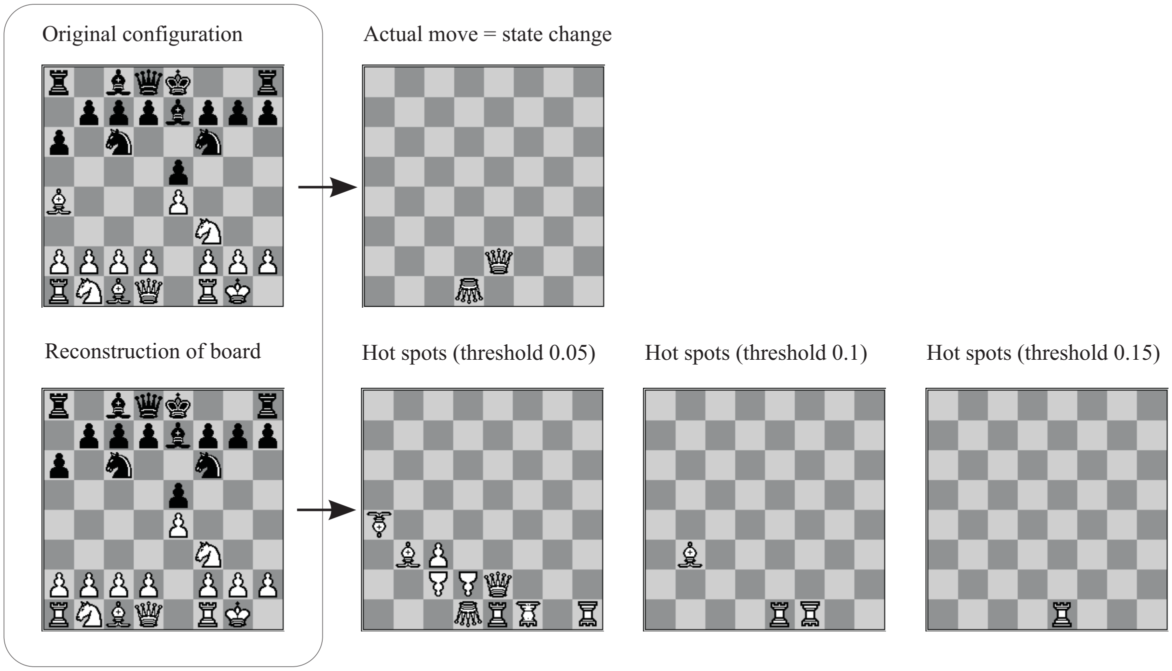

Original configuration
Actual move = state change
Reconstruction of board
Hot spots (threshold 0.05)
Hot spots (threshold 0.1)
Hot spots (threshold 0.15)

Figure 5: An example of how high dimensionality makes it possible to mimic cognitively relevant functionalities. First, when using only static configuration data, it turns out that *chunking* can be studied; second, when the derivatives are also included, it becomes possible to attack the challenges of *attention.* On the leftmost images, the observed chess piece configurations $u(k)$ are presented: On top, there is the outlook of the original board, and on the bottom, there is the reconstruction when using a storage of only 100 numeric chunks that are appropriately stacked on top of each other. In such a typical case, almost all pieces can be correctly recalled (the vector $\hat{u}(k)$ is thresholded so that only pieces with relevance $\hat{u}_j > 0.5$ are shown). The remaining images illustrate the "flow" of the game, or derivative $\frac{du}{dt}(k)$ in the current state k: Again, on top, there is the observed change in the configuration, and on the bottom, there is the estimate, visualized applying three different threshold levels. The pieces upside down denote vanishing pieces. It turns out that the "hot spots" are located in relatively appropriately (even the missing bishop is now there), and, as it turns out, it is indeed the expert-selected move that has a strong representation — even though it is not the winner. Note that the reconstruction is purely asociative, and no check for validity is here carried out, so that some "ghost spots" also exist. On top of the associations, higher-level reasoning is needed to screen the reasonable moves

(category/attribute) number i when perceiving the input. As seen in another perspective, the sparse coded momentary weights $\bar{x}_i$ stand for the cognitivistic notion of *short-term memory,* containing "indices" to *long-term memory* constructs. These LTM constructs are the profiles $\bar{\phi}_i$ expressing the elementary patterns of exhaustion of available activation. This scheme is completely distributed and locally controlled; the computer paradigm with its centralized registers, memory units, and data transfer among them, can be abandoned in this framework.

In the system theoretical perspective, relevant concepts are *attractors* in the data space: In appropriate conditions, when the incoming signals match the model, dynamics ends in the corresponding basin of attraction. Because of the distributed nature of the model, many of the available attractors can be simultaneously active. The nonlinear, multimodal distributions of natural data are thus represented by separate mental substructures, each competing for activation.

The contents of a concept are determined essentially in terms of relations to other constructs, in the simplest case these constructs being *examples.* The uniformity of mental phenomena can be extended outside the nominal concepts: Even the content of *feelings* is determined by prototypical experiences.

3.4 On expertise

One can claim that *expertise* in a specific domain is based on appropriate features or chunks existing in the conceptual space. An expert matches the observations against his mental view, thus compressing the data into domain-oriented representations. Applying this low-dimensional representation, missing variables are "filled in" as the known variables are

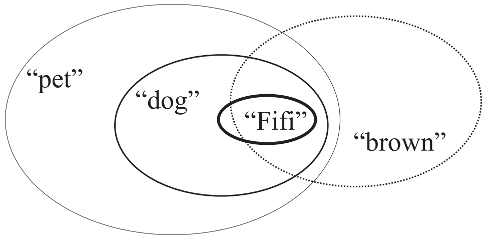


Figure 6: When interpreting the neurons-defined structures in epistemic terms, it can be said that the concept hierarchies become fuzzy: A dog is a subclass of a pet, and Fifi is a subclass of a dog — but, simultaneously, a dog is a part of the contents of a pet, and Fifi is part of dog. Inheritance is not hierarchic but becomes a network: For example, examples of a dog determine what brown color is like, and the concept of brown partly define what dogs are like. Speaking of dogs activates associations to pets, and *vice versa.* Such discussions become less trivial in higher dimensions

matched against the model, and this way, "associative inference" is implemented (see Fig. 7).

The distribution-oriented view of expertise allows subtle, non-binary reasoning, and also merciful degradation of mental capacity as a function of scarcity of resources is manifested. This is visualized, for example, by the chess experiments, where the errors are "expert-like" (again, see Fig. 5).

There is a close connection to case-based reasoning (CBR) here, but now there is some kind of "functional" matching of the mental model against data taking place: All input entities do not have the same significances, but the knowledge structure is taken into account. A solution to the *frame problem* is also reached: The high-dimensional knowledge representations are never isolated from their surroundings, but different kinds of "defaults" are automatically carried along.

4 Contribution of neocybernetics

There are many additional intuitions that are offered by the neocybernetic approach.

Causality. The mapping from u to $\bar{x}$ is valid only if the closed-loop system is stable, that is, the system can affect the environmental variables and balance them. This is not always the case, and analysis of covariance properties of u do not necessarily reveal the system properties. A more fundamental relationship is that between $\bar{x}$ and Δu, the mapping as seen by the system itself. This mapping is interesting because it essentially represents the *effect of the system back onto the environment.*

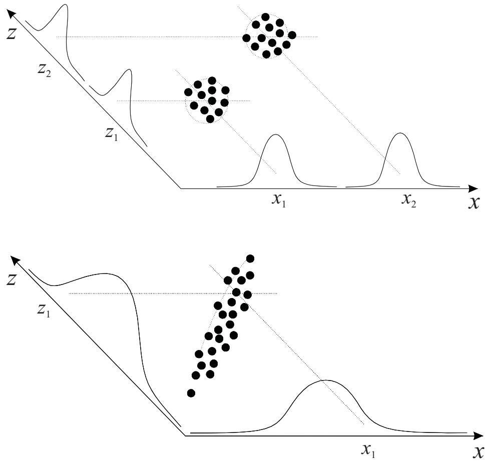


Figure 7: Traditional view of expertise (on top) makes it possible to implement rules of the form IF $x = x_i$ THEN $z = z_i$, etc., whereas when distributions are employed, inference becomes subtler, being an associative (maximum likelihood) pattern matching process against the existing high-dimensional knowledge model

As observed originally by Hume, one cannot ever see *causalities* in data, only *correlations*, that is, one cannot detect cause/effect relationships. But as Kant has said, it is causal structures that are constructed by the human mind. How is this possible then without employing some *a priori* understanding?

It can be claimed that the neocybernetic model offers a solution to this dilemma. Because it is only ones own actions Δu, as induced by the environment, that are being coded in $\bar{x}$, one implicitly knows the structure among causes and effects — there is no paradox any more here. True causality structures are indeed built deep in the Hebbian feedback models.

Consciousness. There are many contradicting intuitions of how consciousness should be defined — the heated controversies being, of course, caused by the fact that consciousness is the essence of our speciality among animals. The views vary from the highest (consciousness is the culmination of intelligence) to the lowest level (consciousness is ability to feel something like pain), or even below that (consciousness can only be explained in terms of quantum effects).

Awareness of *self*, or "knowing that one knows that one knows", is assumedly a holistic, emergent phenomenon that cannot be reduced. However, in the adopted framework this structure of infinite recess can again be collapsed, and a concrete definition for consciousness can be proposed. In the neocybernetic spirit, it can be assumed that the mental machinery constructs a more or less sophisticated model of the environment; when this model becomes complex enough, the “self” becomes a relevant entity in the model that succesfully helps in structuring the observations and behaviors in the environment. When such self emerges as an independent entity in the mental model, there is consciousness. This would mean that animals have consciousness in varying degrees — but also non-biological cybernetic systems would be conscious to some extent. On the other hand, a small child not distinguishing itself from its mother is *not yet* conscious — but the “brain prosthesis” can truly capture the mind.

Intersubjectivity and interobjectivity. In complexity theory that is based on chaos theoretical starting points, it is often claimed that the value of the models is questionable as small deviations in initial conditions result in completely different outcomes. Now, on the other hand, stochastic variations are not significant: It is statistically relevant constructs or attractors of dynamic processes within the phenosphere that are being captured, and role of the transients fades away.

This observation has its effects also on AI, or, indeed, to the *theory of mind* itself: If the modeling principles are the same, reaching towards optimized representations, and if the environments are the same, two different modeling processes end up in essentially the same “world view”. This applies not only to humans that have their subjective experiences — this *intersubjectivity* can be extended also between natural and artificial brains. Constructing “smart machines” in the deep AI sense *is* possible.

One can even extend these considerations beyond the mind, to the principles of modeling in general: The goal of AI is to make the the computer somehow understand — or model — the world. In (Hyötyniemi, 2006b) it is observed that nature itself constructs models — a higher trophic layer that is evolutionarily reasonable tries to capture the behaviors of the lower layer to maximally exploit it. If a human (or computer) then models the same ecosystem (applying appropriate modeling framework), is the artificially-made and nature-made models somehow essentially different?

In modeling theory one is traditionally humble: It is assumed that a model can only capture a shadow of the real-life complexity. But if the real life is also a model or a model-based simulation, *interobjectivity* can be reached, where the man-made model is essentially the *same thing* as the reality itself. Essence of systems is not in the building blocks but in the information structures. Epistemology becomes ontology. It may be that *metaphysics* finally becomes a science.

5 Conclusions

To goal of complex systems thinking is to find simple underlying principles beyond different kinds of complex phenomena. In the neocybernetic perspective, it seems that there truly exist common basis for neural/cognitive systems and other cybernetic ones.

It is not only so that cybernetic understanding can be exploited in AI and analysis of cognitive systems, as was done above — there is also contribution in the inverse way. Perhaps the intuitions from AI can be employed when studying what kind of ontologies there can exist in other cybernetic systems. For example, can the structures emerging in *genetic systems* be studied in terms of similar concepts? What is more — can the genetic code be translated into a human-understandable language?

And the social systems among individual people — it may turn out that extending the cognitive system beyond the limits of a single brain can be understood and analyzed in the neocybernetic framework. Constructing a common “supermodel” among the individuals is all dependent of the information transfer between the minds. Perhaps the idea of EQ or “emotional intelligence quotient” is the IQ of intelligent organizations?

References

W.G. Chase and H.A. Simon. *The mind’s eye in chess*. In *Visual Information Processing* (ed. Chase, W.), Academic Press, New York, 1973.

S. Haykin. *Neural Networks — A Comprehensive Foundation*. Prentice–Hall, Upper Saddle River, New Jersey, 1999.

D.O. Hebb. *The Organization of Behavior: A Neuropsychological Theory*. John Wiley & Sons, New York, 1949.

F. Heylighen and C. Joslyn. *Cybernetics and Second-order Cybernetics*. In R.A. Meyers (ed.). Encyclopedia of Physical Science & Technology (3rd ed.), Academic Press, New York, 2001.

H. Hyötyniemi. *Elastic Systems: Another View at Complexity*. SCAI’06, Helsinki, Finland, 2006a.

H. Hyötyniemi. *Neocybernetics in Biological Systems*. Helsinki University of Technology, Control Engineering Laboratory, 2006b.

J. Laaksonen. *Subspace Classifiers in Recognition of Handwritten Digits*. Acta Polytechnica Mathematica; Mathematics, Computing and Management in Engineering series No. 84, Espoo, 1997.

Proceedings of the Ninth Scandinavian Conference on
Artificial Intelligence (SCAI 2006)

Espoo, Finland, October 2006

Publications of the Finnish Artificial Intelligence Society 22

ISBN-13: 978-952-5677-00-3 (paperback)
ISBN-10: 952-5677-00-1 (paperback)
ISSN 1238-4658 (Print)

ISBN-13: 978-952-5677-01-0 (PDF)
ISBN-10: 952-5677-01-X (PDF)
ISSN 1796-623X (Online)

Otamedia Oy

Additional copies available from:

Finnish Artificial Intelligence society FAIS
Secretary Susanna Koskinen
Henrikintie 7 A
00370 Helsinki
Finland
office@stes.fi
http://www.stes.fi/

Table of Contents

Analysing Interdisciplinarity

Genetic Algorithms and Computer Vision

Machine Learning

Robotics and Agents

Digital Games

Elastic Systems

Foreword

The first SCAI was started in 1988 in Tromsø, Norway. It has since then been held in Nordic countries. Although its primary function is to bring together researchers from the Nordic countries, it has constantly attracted a rather international participation.

There have been strong connections between Nordic countries for long time in the history. In the modern times, those connections have provided the citizens, companies and researchers many kinds of opportunities. As one practical example, one can mention the Nordic Passport Union, created already in 1954, well before Schengen agreement. It allowed citizens of Denmark, Finland, Iceland, Norway and Sweden to cross border districts without carrying and having their passport checked. Moreover, technological developments have been boosted by the possibility of bringing new products to this highly developed area. A primary example is the the Nordic Mobile Telephony (NMT), the world's very first multinational cellular network in 1981. The NMT was later on introduced in other countries, making ground for the GSM that became a widely adopted international standard. In the area of information technologies, many innovations and successful concepts and products stem from the Nordic countries including Linux, MySQL, IRC and Skype. Alongside with these Finnish and Swedish examples, one can mention that for the past two years Denmark has ranked first on the Economist Intelligence Unit's e-readiness list that indicate how amenable a market is to Internet-based opportunities. Moreover, Norway has played since 1990 a pioneering role in applying telemedicine in healthcare services. These kind of examples related to the IT area in general are an important basis for successful research and education in the area of artificial intelligence. As the representatives of the Finnish hosts, we are pleased to mention two excellent local examples. Professor Tuomas Sandholm, an alumni of Helsinki University of Technology, was the 2003 recipient of the Computers and Thought Award, given every two years by the International Joint Conference on Artificial Intelligence (IJCAI) to an outstanding young scientist. The award is considered to be the premier prize for AI researchers under the age of 35. Academician Teuvo Kohonen has been highly successful for decedes in an area that was for quite some time neglected in the AI community but has later approved to be very useful in many tasks and applications. Many of his innovations are central components in areas such as pattern recognition, data mining, robotics, knowledge representation and statistical machine learning. This year, we are pleased to be able to welcome Teuvo Kohonen to open SCAI 2006 as the honorary chair of the conference.

We are grateful to all the active researchers who have submitted their contribution to the conference. The conference programme is divided into six thematic sessions named "Analyzing Interdisciplinarity", "Genetic Algorithms and Computer Vision", "Elastic Systems", "Machine Learning", "Robotics and Agents", and "Digital Games". We thank all the session chairs who have kindly agreed to provide an introduction to each of the thematic areas. Parallel to the SCAI 2006 conference, the Finnish AI Conference (STeP 2006) is organized with two main themes, "Semantic Web at Work", and "Artificial Intelligence". Professor Eero Hyvönen’s efforts to ensure the success of the STeP 2006 conference we acknowledge gratefully.

We are also grateful to Professor Tom Ziemke (Skövde) and Nokia Research Fellow Ora Lassila (Cambridge) who kindly accepted the invitation to give a keynote talk in the conference. The talks are based on their outstanding research. Ora Lassila is an acknowledged expert in the area of Semantic Web, the core ideas of which were described in a Scientific American article he coauthored with Tim Berners-Lee and James Hendler in 2001. Tom Ziemke's succesful research covers several important and interrelated topics such as cognitive, evolutionary and adaptive robotics, and embodied, enactive and distributed cognition.

We wish to warmly thank the international programme committee whose active role has been instrumental for the success of the conference. The programme committee members both helped in specifying the conference themes and, in particular, ensured a timely and thorough evaluation of the papers. We express our gratefulness to all the members of the organizing committee for their efforts.

We are grateful to the organizations and companies involved: Finnish Artificial Intelligence Society, Helsinki University of Technology, city of Espoo, Intopii, Nokia, and European Coordinating Committee for Artificial Intelligence.

We wish that the SCAI 2006 conference will be remembered as a useful scientific event and as a memorable gathering in which old friends met and new friends were made.

Espoo, 29th of September, 2006

Timo Honkela, Tapani Raiko, Jukka Kortela, and Harri Valpola

Committees

Honorary Chair

Teuvo Kohonen, Helsinki University of Technology

Program Committee

Finland

Timo Honkela, Helsinki University of Technology (Chair)
Erkki Oja, Helsinki University of Technology
Eero Hyvönen, Helsinki University of Technology
Jarmo Alander, University of Vaasa
Pekka Ala-Siuru, Technical Research Centre of Finland (VTT)
Tomi A. Pasanen, University of Helsinki
Samuel Kaski, Helsinki University of Technology
Ville Kyrki, Lappeenranta University of Technology
Tapani Raiko, Helsinki University of Technology
Pentti Haikonen, Nokia Research Center
Harri Valpola, Helsinki University of Technology
Iina Tarnanen, Helsinki University of Technology
Krista Lagus, Helsinki University of Technology
Otto Lappi, University of Helsinki
Jukka Kortela, Artificial Intelligence Technologies, University of Oulu
Matti Miettinen, RD-Tech
Janne Heikkilä, University of Oulu
Juha Röning, University of Oulu
Heikki Hyötyniemi, Helsinki University of Technology

Norway

Bjørnar Tessem, University of Bergen
Weiqin Chen, University of Bergen
Richard E. Moe, University of Bergen
Terje Kristensen, Bergen Univesity College
Keith L. Downing, Norwegian University of Science and Technology
Tore Amble, Norwegian University of Science and Technology
Helge Langseth, Norwegian University of Science and Technology
Magnus Lie Hetland, Norwegian University of Science and Technology
Agnar Aamodt, Norwegian University of Science and Technology

Denmark

Brian Mayoh, University of Aarhus
Henrik Hautop Lund, University of Southern Denmark
John Hallam, University of Southern Denmark
John Gallagher, Roskilde University
Gregers Koch, University of Copenhagen
Thomas Bolander, Technical University of Denmark
Manfred Jaeger, Aalborg University

Sweden

Peter Funk, Mälardalen University
Andrzej Szalas, University of Linköping
Stefan Johansson, Blekinge Institute of Technology
Thorsteinn Rögnvaldsson, Halmstad University
Lars Karlsson, Örebro University
Fredrik Heintz, Linköping University

Organising Committee

Tapani Raiko, Helsinki University of Technology (Chair)
Pentti Haikonen, Nokia Research Center
Harri Valpola, Helsinki University of Technology
Iina Tarnanen, Helsinki University of Technology
Krista Lagus, Helsinki University of Technology
Otto Lappi, University of Helsinki
Jukka Kortela, Artificial Intelligence Technologies, University of Oulu
Matti Miettinen, RD-Tech
Matti Pöllä, Helsinki University of Technology

Cover Art

Jussi Timonen

Author Index

Making Things Think.

INTOPII

WWW.INTOPII.FI

ARTIFICIALINTELLIGENCE.FI