JYVÄSKYLÄ STUDIES IN COMPUTING 23

Anssi Lensu

Computationally Intelligent Methods for Qualitative Data Analysis

Esitetään Jyväskylän yliopiston informaatioteknologian tiedekunnan suostumuksella
julkisesti tarkastettavaksi yliopiston Villa Ranan Paulaharju-salissa
joulukuun 21. päivänä 2002 kello 12.

Academic dissertation to be publicly discussed, by permission of
the Faculty of Information Technology of the University of Jyväskylä,
in the Building Villa Rana, Paulaharju Hall, on December 21, 2002 at 12 o'clock noon.

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2002

Computationally Intelligent Methods for Qualitative Data Analysis

JYVÄSKYLÄ STUDIES IN COMPUTING 23

Anssi Lensu

Computationally Intelligent Methods for Qualitative Data Analysis

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2002

Editors
Tommi Kärkkäinen
Department of Mathematical Information Technology, University of Jyväskylä
Pekka Olsbo, Marja-Leena Tynkkynen
Publishing Unit, University Library of Jyväskylä

URN:ISBN 9513913554
ISBN 951-39-1355-4 (PDF)

ISBN 951-39-1374-0 (Nid.)
ISSN 1456-5390

ABSTRACT

Lensu, Anssi
Computationally Intelligent Methods for Qualitative Data Analysis
Jyväskylä: University of Jyväskylä, 2002, 57 p. (+included articles)
(Jyväskylä Studies in Computing
ISSN 1456-5390; 23)
ISBN 951-39-1374-0 (nid.) 951-39-1355-4 (PDF)
Finnish summary
Diss.

This study focuses on computationally intelligent methods, which are applied to the analysis of survey data in educational research. The methods can be used with complex data sets, which contain several data types. Each data type is analyzed in a separate subanalysis, and the results from these subanalyses can be combined. The methodology makes it possible to locate groups of similar answers from the subanalyses, and to identify these groups using background information. It also allows one to compare groups that are selected from different subanalyses, from different populations, and to locate and identify similar textual answers. In connection to this study, a software application has been created to test the developed methods.

Keywords: exploratory data analysis, data mining, knowledge discovery, self-organizing maps, model selection, information theory, parallel computing

Author Anssi Lensu
Department of Mathematical Information Technology
University of Jyväskylä
Finland

Supervisor DrTech Pasi Koikkalainen
Department of Mathematical Information Technology
University of Jyväskylä
Finland

Reviewers PhD Timo Honkela, Chief Scientist
Gurusoft Oy
Helsinki
Finland

PhD Petri Myllymäki, Docent
Department of Computer Science
University of Helsinki
Finland

Opponent Professor Henry Tirri
Department of Computer Science
University of Helsinki
Finland

ACKNOWLEDGMENTS

I am very grateful to DrTech Pasi Koikkalainen who has guided me and my research for the past 6 years. He has given me a lot of insight about the field of computer science, and he has also co-authored most of my work. I want to express my thanks to all my co-workers in the Laboratory of Data Analysis and on the Department of Mathematical Information Technology of the University of Jyväskylä. I especially wish to mention PhD Erkki Häkkinen and PhLic Jouni Raitamäki with whom the studying and learning have been enjoyable. I have also learned a lot about statistics from Prof. Antti Penttinen and about the theory of algorithms from Prof. Pekka Orponen. In addition, the seminars organized by the Center for Mathematical and Computational Modeling (CMCM) have been quite educational.

I am also grateful to the reviewers, PhD Timo Honkela and PhD Petri Myllymäki, who gave me quite valuable comments about my work. I also wish to mention Prof. Päivi Häkkinen who checked the part related to qualitative research, and MSc Steve Legrand who was responsible for the proofreading. I also want to thank my co-workers at the Institute for Educational Research, especially Prof. Pirjo Linnakylä, Prof. Päivi Häkkinen, and MSc Antero Malin to name a few, for presenting their research problems and helping me with the area of educational research. They have also provided a very good audience for whom I could present my ideas.

This research work was supported by the Academy of Finland under project #37190, CATO/LAMDA; the Graduate School of Computing and Mathematical Sciences at the University of Jyväskylä, COMAS; and the National Technology Agency, TEKES, under projects DAEMON and IDEAL. I also wish to thank our educational and industrial partners, the Institute for Educational Research of the University of Jyväskylä and Sonera Corporation, for allowing me to use their data, and the Jyväskylä Science Park, JSP, for their funding for our Laboratory of Data Analysis.

I wish to express my deepest gratitude to my wife Anne-Maija for being so patient and supportive, and to my daughter Linnea for bringing so much joy to our family. My parents, Leena and Pertti, and my brother Lasse also deserve special acknowledgment. Without my mother's parental guidance, my father's genes (he died when I was four years old), and my brother's good example (and the inspiring discussions with him), I probably would never have made it this far.

Jyväskylä, December 4, 2002

Anssi Lensu

CONTENTS

LIST OF INCLUDED ARTICLES

[A] Lensu, A. and Koikkalainen, P.
'Analysis of Gallup Data through the Neural Data Analysis Environment'.
In *Proc. HELNET'97: Workshop on Neural Networks*.
Montreux, Switzerland, 1997.
(There is no ISBN number for the publication.)

[B] Lensu, A. and Koikkalainen, P.
'Analysis of Multi-Choice Questionnaires through Self-Organizing Maps'.
In *Proc. ICANN'98:* 8^{th} *International Conference on Artificial Neural Networks, Vol. 1*.
Pages 305–310. Springer-Verlag, London, 1998.

[C] Lensu, A. and Koikkalainen, P.
'Analysis of Gallup Questionnaires through Self-Organizing Maps'.
In *Proc. STeP'98:* 8^{th} *Finnish AI Conference*.
Pages 171–180. Finnish Artificial Intelligence Society, Helsinki, 1998.

[D] Lensu, A. and Koikkalainen, P.
'Similar Document Detection using Self-Organizing Maps'.
In *Proc. KES'99:* 3^{rd} *International Conference on Knowledge-Based Intelligent Information Engineering Systems*.
Pages 174–177. IEEE Press, Piscataway, NJ, 1999.

[E] Lensu, A. and Koikkalainen, P.
'Computationally Intelligent Methods for Educational Research'.
In *Perspectives on the Age of the Information Society*.
Pages 125–140. Tampere University Press, 2002.

[F] Lensu, A. and Koikkalainen, P.
'A Parallel Implementation of the Tree-Structured Self-Organizing Map'.
In *Proc. PARA 2002:* 6^{th} *International Conference on Applied Parallel Computing*.
Pages 370–379. Springer-Verlag, Berlin Heidelberg, 2002.

[G] Lensu, A. and Koikkalainen, P.
'Complexity Selection of the Self-Organizing Map'.
In *Proc. ICANN 2002: International Conference on Artificial Neural Networks*.
Pages 927–932. Springer-Verlag, Berlin Heidelberg, 2002.

[H] Lensu, A. and Koikkalainen, P.
'A Unified System to Analyze Heterogeneous Survey Data Sets'.
(Manuscript)

[I] Lensu, A. and Koikkalainen, P.
'Analyzing Survey Data with the Neural Data Analysis Environment'.
Accepted to STeP 2002: Finnish AI Conference.
December 16–17, 2002, Oulu, Finland.
To be published.

LIST OF SYMBOLS

General symbols used throughout the thesis	
$\#A$	Evaluates the number of items in set A
d_v	Dimensionality of a model used for describing data
d_x	Dimensionality of the data Ω
$\mathbb{E}$	Expectation
$\mathcal{G}$	Categorical scale
N_v	Number of clusters (or neurons in a SOM)
N_x	Number of data points in Ω
Ω	Data set to be analyzed
Ω_k	Cluster of similar data items
x	Single variable
$\boldsymbol{x}(j)$	Data vector of Ω
$\mathcal{Y}$	Set of features representing all answers of a person

Self-Organizing Map and its evaluation	
$\alpha(t)$	Training speed function
$c(\cdot)$	Function to choose the closest neuron
E	Value of the SOM potential function
$\boldsymbol{\varepsilon}$	Representation error of a principal surface
$h_{c,k}$	Value of the neighborhood kernel function
ℓ	TS-SOM layer containing $2^{d_v \ell}$ neurons
$\sigma(t)$	Neighborhood size or kernel width function
$\boldsymbol{v}$	Point on a principal surface
$\hat{\boldsymbol{v}}(k)$	Neuron prototype k
$\boldsymbol{w}(k)$	d_x-dimensional weight vector of neuron k
$\boldsymbol{x}(\boldsymbol{v})$	Principal surface in data space

Fuzzy Set Theory	
$\mu_{\tilde{s}}(x)$	Membership function for fuzzy set $\tilde{s}$
$\tilde{s}$	Fuzzy set

Statistical Natural Language Processing	
c_i	Character within text
d_j	One document from corpus
$\boldsymbol{d}(j)$	Index term vector for document d_j
κ	Number of previous conditioning characters
n	Index of the word to be predicted (n-gram model)
N_t	Number of terms in $\mathcal{T}$
ω_i	Word within a document
$\boldsymbol{q}$	Index term vector for the query
$\mathcal{T}$	Set of index terms
t_i	One term of the set $\mathcal{T}$
$w_i(j)$	Weight for term t_i within document d_j

Model selection	
$C(\cdot)$	Coding function
$\boldsymbol{\epsilon}(j)$	Residual for data vector $\boldsymbol{x}(j)$
$f(\cdot)$	True model or distribution
$f_H(\cdot)$	Parametric histogram density
$f_G(\cdot)$	Gaussian kernel function
$g(\cdot)$	Candidate model or distribution
$H(\cdot)$	Function to calculate the Shannon's entropy of a probability distribution
$I_{KL}(\cdot)$	Kullback-Leibler distance (or divergence)
$I_{SC}(\cdot)$	Stochastic complexity of parametrized data
K	Number of estimated parameters in the model
$\mathcal{L}(\cdot)$	Likelihood of model given data
$L(\cdot)$	Code length evaluation function
$\mathcal{M}$	Model family or set of models
m_i	One chosen model from model family $\mathcal{M}$
N_m	Number of histogram pins or kernel functions
$p(x)$	Probability density function for a discrete x
$\boldsymbol{\theta}$	Parameter vector for candidate model g
$\mathcal{X}$	Set of possible values for X
X	Random variable, which takes values in $\mathcal{X}$
y	Integration variable

1 INTRODUCTION

> Once upon a time, statisticians only explored.
> Then they learned to confirm exactly –
> to confirm a few things exactly,
> each under very specific circumstances.
> As they emphasized exact confirmation,
> their techniques inevitably became less flexible.
>
> *John Tukey*

The field of educational research contains a large number of research problems, which are typically solved using either *qualitative research* methods (Tesch 1990, Miles and Huberman 1994) or traditional parametric statistical analyses, depending on the type of the data. The problems with qualitative methods are that they require a lot of human work, and that (normally) their results are not quantitatively measurable. And even though the parametric statistical methods are used in qualitative data analysis to provide exact measurable results, they depend on the chosen hypotheses. Since all possible hypotheses cannot be tested, the interesting and essential contents of the data set may be left unnoticed. Other key problems with the parametric statistical methods are that they include strict assumptions regarding the distribution of the data, and that their applicability is limited to cases, where the number of variables is low. Therefore, there is a need for computer assisted analysis methods, which could be used with both quantitative and qualitative data. These methods are needed in order to reveal the intrinsic similarities between the multivariate or otherwise complex data items.

Educational research data sets may contain several data types, which all describe the characteristics or opinions of the human beings. For example, surveys contain questions, to which the answers are given in numerical, categorical or textual form. In addition, the number of questions and the number of respondents may be large, which increases the work needed for analyzing the data set manually. Therefore, the analysis methods need to be able to efficiently make a compact representation of the contents of the data even though the data set may be polymorphic, complex and large. To be able to get a condensed idea of the opinions of real people and their *collaborative* (Dillenbourg 1999) behavior, the groups of data items, which are similar to each other, can be located and identified. These groups can then be studied instead of the individuals to find the underlying reasons, which make the group representatives different from the rest of the survey sample. Other questions about the contents of the data can also be answered more easily using the located groups.

The *Exploratory Data Analysis* (EDA) (Tukey 1962, Tukey 1977), and *data mining* and *Knowledge Discovery in Databases* (KDD) (Frawley et al. 1992, Fayyad 1996, Fayyad et al. 1996) methods that are presented in this work are data-driven.

These methods avoid making prior assumptions on what kind of data items should be found or what the data distribution is. Instead, they try to locate similarities from the multivariate data sets using computational methods, such as *unsupervised statistical learning* (Hastie et al. 2001). However, to be able to choose the correct preprocessing methods and to make the results more clear and motivated, the prior knowledge of the researcher has to be used to guide the analysis.

Typically, the role of the data analysis is the developing of qualitative understanding of the contents of data. Exploratory data analysis can be used to the analysis of qualitative data by developing suitable preprocessing methods. The EDA or data mining methods can be used in connection with qualitative research methods, such as *grounded theory* (Glaser and Strauss 1968) or *transcendental realism* (Miles and Huberman 1994). The interactive nature of the computer assisted methods also allows the researcher to backtrack to an earlier phase of the analysis easily, if the results are not satisfactory. Therefore, the researcher may get a more thorough insight into the data than is possible with traditional methods.

1.1 Studied Educational Research Problems

We have concentrated on problems where data has been difficult to analyze using traditional statistical methods, or has typically required a lot of manual work. Survey data with many data types and lots of variables are hard problems for educational researchers. In addition to textual answers, they usually contain multiple-choice questions, as well as categorical and numerical background information.

This study does not address the problems related to survey planning, such as questionnaire design, or *sampling*. The design of questionnaires is strongly related to the field of research, and thus general methods can only be developed for the post evaluation of answering consistency. A general theory of sampling using unequal probabilities and probability-weighted estimation was suggested in (Horvitz and Thompson 1952). Using that theory, the located results of data analysis can be extended to reflect the situation in the whole population.

One good example of a complex data set is the survey (called Peruskoulun arviointi 1995) conducted by the Institute for Educational Research of the University of Jyväskylä in 1995. In this survey a questionnaire containing 29 multiple-choice questions, 16 free-form textual questions, and some numerical and categorical background information was used to assess school satisfaction in Finnish comprehensive schools. The multiple-choice questions (or actually statements) were designed to be responded to on a 4-point agree-disagree scale $\mathcal{G}_1 = \{definitely\ disagree, mostly\ disagree, mostly\ agree, definitely\ agree\}$, which is an *ordered categorical* or *ordinal* scale. The textual questions contained variable length answers, which were only about one to three sentences long. The background information *was not supposed to be used* for locating similar items. Instead, it was to be later utilized for the identification of the located groups of people.

The sample size was 1380 students from different parts of Finland.

The multiple-choice questions (Williams and Batten 1981) were designed to measure students' feelings about school using Quality of School Life as the perspective. They have been used to evaluate the quality of school life in at least 30 countries around the world. For example, in 1991 the questions were used in connection with the IEA (International Association for the Evaluation of Educational Achievement) International Reading Literacy Study, where the sample size was more than 100 000 students. Williams and Roey summarize the design process in (Williams and Roey 1997) and explain why the questions have been divided into six semantic categories: General affect, Negative affect, (Achievement and) Opportunity, Teachers, Identity, and Status. *Factor analysis* (Thurstone 1949) has been used to show that these categories (or factors) mostly agree with their theoretical design. The textual questions also measured the same kind of aspects of school life as the multiple-choice questions, and were designed by the Finnish researchers.

Regarding these data sets, the educational researchers have been interested in at least the following research questions (Linnakylä 1996, Linnakylä and Brunell 1997, Linnakylä and Malin 1997, Linnakylä and Malin 1998, Malin and Linnakylä 2001):

- What are the background factors affecting school satisfaction?
- What kind of groups of similar individuals can be found from the whole survey sample if only their answers to the presented questions are studied?
- How do the answers of the individuals in some located group differ from the answers given by the rest of the survey sample?
- If a group of unsatisfied students is found, what are the underlying reasons, which make them negative towards school?
- Do negative teacher – student relations cause negative attitude towards school in general?
- How do survey samples from different populations, for example from different countries, differ from each other?
- How have the school satisfaction and teacher – student relations changed in the Finnish population from 1991 to 1995?
- How could exact meaning be extracted from inexact qualitative data?
- Do the textual answers support the opinions obtained from the multiple-choice answers within one semantic category?

These questions can be reformulated into a few data analysis or data mining questions:

- How can groups of similar data items (answers) be located?
- What background information is available about these groups?
- How can the correlations between different parts of data be evaluated and represented?

- How can different data sets with several semantic aspects be compared?
- How can textual data be analyzed and clustered?

1.2 Data Analysis Problems

It would be almost impossible to directly and automatically convert the textual and the categorical answers into commensurable parts of a *feature vector*, which would represent all the answers of a person. Using a lot of manual work including reading of all textual answers many times and analyzing the quantitative variables with statistical methods, a set of features $\mathcal{Y}$ representing the contents of the answers probably could be found. In the included articles we propose an interactive system where the different data types and theoretical aspects of the questionnaire are analyzed separately.

If the number of multiple-choice questions is large, locating similar responses becomes difficult due to the large number of possible value combinations. The use of clustering methods becomes difficult due to the *curse of dimensionality* (Bishop 1995, Duda et al. 2001, Hastie et al. 2001), and because it is hard to decide on a suitable number of clusters if the responses are about evenly distributed in the space of possible answer combinations. This problem can also be formulated another way: Which data items should be considered similar and which different, if no distinct clusters can be found?

Free-form textual data is complex due to the almost infinite number of possibilities how the letters of some alphabet $\mathcal{A}$ can be combined into variable length words. The words are further used to form variable length sentences, which integrate into paragraphs and documents. Even though the letters cannot be freely combined in natural languages, the size of the dictionary is always huge, and new words are introduced frequently. In *agglutinative* languages, such as Finnish, prefixes and suffixes are used to create *inflections* and *derivations* (*morphological* forms) from the base words, and several words can even be combined into compound words. There could also be locally used *dialects*, which affect the spelling of some words.

The same kind of versatility applies also to words within sentences, because the word order need not be strict and changes in the order may affect the meaning of the sentences. The meaning may be more severely affected by a simple negation. However, there are application areas, in which the text data has more structure and similar expressions occur frequently. For example, answers to direct questions typically do not contain as poetic language as some other text *corpora*, and can therefore be analyzed more easily.

1.3 Other Application Areas for Presented Methods

Surveys and censuses are conducted by national statistical bureaus, statistical departments of cities, universities, companies, and so on, all the time. If the ques-

tionnaires are not simple, efficient data analysis methods are needed. Textual data are used almost everywhere, and therefore efficient filtering and classification methods are needed a lot. Typical application areas include querying the World Wide Web, Email filtering to reduce the amount of spam, searching the library catalogs, and so on.

Typically the problem is to build useful description vectors for the documents to be queried, filtered or ordered, and for the query if applicable, and then somehow classify whether each document is relevant or not. Rule based systems, which are created using domain knowledge, may be efficient for some specific data sets, see the discussion in (Sebastiani 2002), but they have to be created and updated by a true expert of the domain. However, model based systems, which are created using statistical learning, can be built without any expert knowledge, but then a lot of data is needed for the training.

In addition to surveys, we have applied our textual data analysis methods also for locating similar data communication problem descriptions using real-world data from the Sonera Corporation. The goal was to develop a user interface with which a large text database could be queried using a new problem description, and the solutions to previously occurred similar problems could be obtained. This has been briefly referred to in Article [D].

1.4 About the Content of the Preface

The studied problem types and typical contents of the data sets were first presented in Chapter 1.1. Next, a review of the data analysis and modeling methodology behind the methods used in the articles is presented in Chapter 2. The unsupervised learning methods have to be regularized to avoid over-fitting to data. An idea how this can be done is presented in Chapter 3, which contains a review of model selection and assessment, and describes our approach for the Tree-Structured Self-Organizing Map (TS-SOM). Finally, a parallel TS-SOM training method for large data sets (often obtained from document collections) is introduced in Chapter 4.

2 DATA ANALYSIS METHODOLOGY

This chapter presents a literature review of some of the methods used for data analysis. The methodology has been chosen to support the arguments given in the included articles. Qualitative research methods are introduced to give insight about their strengths and weaknesses, and to point out possibilities of computer programs in this context. However, our approach is to use exploratory data analysis methods for data modeling and visualization. In exploratory data analysis unsupervised learning methods can be used to assist exploration. Our methods are mostly based on the Self-Organizing Map and its use for the visualization of data and models. Methods for the preprocessing of categorical and textual data are also introduced by presenting the idea of fuzzy sets and natural language processing methods.

2.1 Qualitative Research

> Strictly speaking, there is no such thing as qualitative research.
> There are only qualitative *data*.
>
> *Renata Tesch*

Qualitative research is still a term used by many scholars in human sciences including sociology, psychology, and education (Tesch 1990). It is used while referring to the process of knowledge production from qualitative data. Qualitative data usually means descriptive information, for which exact and unambiguous distance metrics cannot be defined. Typically, it can be words, but also pictures, video clips, and music could be thought to be qualitative. In this thesis, only textual and categorical (or *discrete*) data are considered and studied. Statisticians often regard categorical data as qualitative (Hastie et al. 2001), which can be motivated by noting the limitations of numeric representation.

The categorical variables, which take one value from a set of N_c possible attribute values, $\mathcal{G} = \{value_1, value_2, \ldots, value_{N_c}\}$, are in many cases represented with numbers $\mathcal{G}' = \{1, 2, \ldots, N_c\}$ in computers. However, this does not always indicate that there would be an order between the choices. The reason to use numbers is just that numbers are more economical in both for the storage and for the comparison of categories. *Ordered categorical* or *ordinal* variables use a scale, on which there exists an order between the choices, for example $value_1 < value_2 < \cdots < value_{N_c}$, but still the numbers need not indicate the "magnitude" of the choices, or the distances between adjacent choices need not be uniform. People may also disagree on how much each choice differs from the others quantitatively.

Because there is a clear structure in categorical data, it is usually not analyzed using qualitative research methods. Instead, the analysis of categorical data is usually considered as a statistical problem. The meaning of certain categorical

answers or their combinations can be evaluated after some statistical analysis method, such as clustering, is applied.

The ordering of textual expressions is much more difficult. Usually only quite similar textual expressions can be located automatically. A human expert is the only real judge, who is able to say if two completely different sentences mean the same thing in the real world. Therefore, computer systems are only able to notice similarities in the texts and point them out to the user. Qualitative research methods are able to continue the analysis from there.

2.1.1 Qualitative Analysis Types

For the analysis of textual data Tesch presents a comprehensive taxonomy (Tesch 1990). The study or exploration of language is divided into two main categories: study of language *as a structure* and its study *as communication*. Linguistics and anthropology belong to the first category. Linguists are mostly interested in the syntax of sentences, the morphological forms of the words, ambiguity of expressions, exact semantics of words, and so on. And anthropologists study the cultural aspects, such as what meaning a word has in a culture, and how other words relate to it. Recently, the cultural aspects have also become more important in the other fields of study. Social sciences, especially educational research, study the second category of communicational aspects. *Literary criticism* (humanities) and *interpretation of text* (hermeneutics) also study communicational aspects, but the goals are quite different. Figure 1 illustrates the different tasks of exploration of language, and how they relate to each other.

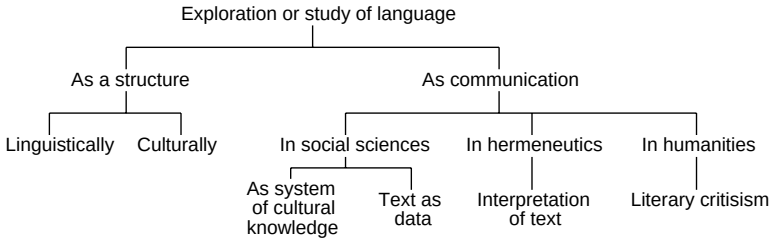

FIGURE 1: The high level taxonomy of exploration of language adopted from (Tesch 1990) p. 57.

In educational research the textual information is in many cases seen as data, and *content analysis* or *discourse analysis* is done, depending on what kind of study the data relates to. If the communication is seen as a *process*, for example as in a conversation, discourse analysis can reveal how language is used in certain situations or how the presence of other people affect the communication. However, the survey data only contains static content and therefore content analysis accompanied by *discovery of regularities* is the most reasonable approach. The central ideas in content analysis are to categorize words or sentences into a few groups, which

are relevant for the research purpose, or to make inventories of how frequently each word is used, or to explore in what kind of contexts the words have been used. Figure 2 illustrates how the different tasks of qualitative research relate to each other. The approach depends on the research interest, which can also be *comprehension of the meaning of text/action* or *reflection*, which have been omitted from the figure.

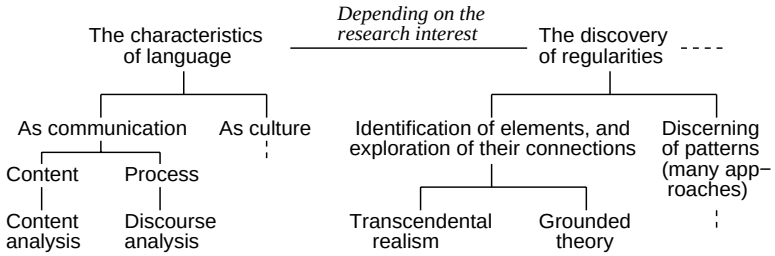

FIGURE 2: The hierarchy of qualitative research adopted from (Tesch 1990) pp. 59, 61 and 63.

Traditionally, the discovery of regularities from a *text corpus* is done by reading texts through many times, categorizing similar textual expressions, and presenting what kind of relations exist between them. According to Tesch there are two main approaches: *grounded theory* (Glaser and Strauss 1968), and *transcendental realism* (Miles and Huberman 1994). The term transcendental realism was used by Tesch, because the qualitative data analysis methods used by Miles and Huberman were not given a specific name in the first edition of their book in 1984. However, in the second edition Miles and Huberman themselves indicated that *'We see ourselves in the lineage of "transcendental realism" '*, which means that they agree that social phenomena exist also in the objective world, not just in the mind.

Glaser and Strauss motivate grounded theory by claiming that qualitative data has been used in nonsystematic and nonrigorous ways before the World War II, and that monographs based on qualitative data contained just lengthy explanations of results instead of theory. They thus suggested that the generation of theory by comparative analysis could be used in order to make qualitative analysis more systematic. The main idea and the working scheme of grounded theory are clearly illustrated in (Hutchinson 1988). Using the full potential of the method requires that data collection, coding, and analysis are done simultaneously in such a way that focus can be changed, if needed. The goal is to discover theory by sorting incidents found in textual data into categories, which are interpreted by listing their properties, and comparing the categories at the same time until the most coherent categories with clear properties stand out. There are three levels of coding: i) breaking data into pieces and abstracting them, ii) condensing the resulting codes into categories, and iii) deriving theoretical constructs for categories using academic and empirical knowledge. The constant comparing of data and interpretations assisted by memoing of ideas, sampling from data, and

sorting of codes are supposed to help the researcher achieve a sense of closure, which is called saturation.

In the work by Miles and Huberman the aim is to build a qualitative description of the data using matrices and *causal networks*, which *'pull together independent and dependent variables and their relationships into a coherent picture'* (Miles and Huberman 1994). In their analysis model there are three concurrent flows of activity. These flows, data reduction, display, and conclusion drawing/verification, are active already when the data is being collected, and they are continued in the post-collection period. Several methods are suggested for data collection, reduction, visualization, and conclusion drawing.

Both of these analysis processes can be assisted by text data base managers and text retrievers (Tesch 1990). The categorization or data reduction is also a key part in both methods, and therefore a statistical learning method, which would be able to detect similar content from text, could be quite useful for exploration, and for the identification of discoveries. The qualitative research methods can really dig out deep knowledge from the text. However, the results typically are qualitative the same way as the source data.

2.2 Exploratory Data Analysis and Data Mining

In Exploratory Data Analysis (EDA) (Tukey 1962, Tukey 1977) the idea is to avoid making prior assumptions about the contents or distributions of the data. Instead, statistical inferences are made by first conducting an exploratory phase, which is mostly data driven. The goal of this phase is to reveal the intrinsic relations within the data and present them in a form that can be understood easily. The results can be later verified in a confirmatory phase.

Modern EDA methods can be used for Knowledge Discovery in Databases (KDD) (Frawley et al. 1992, Fayyad 1996, Fayyad et al. 1996). In this field the goal is to discover frequently occurring or novel patterns from data. Typically, KDD is seen as a process with many phases including data selection, preprocessing, transformations, data mining, and interpretation and evaluation of the obtained results. Since the terminology is not quite stabilized, yet, people may use the term data mining also to mean the whole KDD process. For the data mining phase, several methods can be used including pattern recognition (Bishop 1995, Duda et al. 2001) and statistical learning (Hastie et al. 2001). In our view, illustrative visualization of the results and evaluation of the data mining process (in addition to evaluation of the results) should be included due to the reasons presented in Chapters 2.7 and 3.

2.3 Unsupervised Statistical Learning

The unsupervised statistical learning methods (Hastie et al. 2001) are well suited for data analysis and mining, because in them only the input features themselves

are observed. No measurements of outputs or responses, such as physical outputs of a process or classification information for the inputs, are needed. These methods can be adapted to each situation by developing suitable *feature extraction* or data *preprocessing* methods. Statistical information about the data is needed for the preprocessing phase, and prior knowledge can be of substantial use, but other information is not necessary for the analysis. The unsupervised learning methods are fully capable of handling multivariate data, because they either *cluster* similar items together or *project* the high-dimensional data onto a lower dimensional latent space.

Clustering methods (Anderberg 1973) attempt to partition data Ω into pointwise separate groups of items, *clusters* Ω_k, within which items are pairwise more similar than items which have been taken from different clusters. If the goal is to simplify the representation, the number of clusters, N_v, has to be less than the number of data items, N_x. Other goals may include the representation of relations between individual data points using a hierarchy. Clustering can be seen as a hard problem, because the number of ways how N_x observations can be sorted into N_v clusters is a Stirling number (Anderberg 1973) of second order

$$\mathcal{S}_{N_x}(N_v) = \frac{1}{N_v} \sum_{k=0}^{N_v} (-1)^{N_v - k} \binom{N_v}{k} k^{N_x} \ . \tag{1}$$

So, for the small problem of assigning 25 observations to 5 groups, $\mathcal{S}_{25}(5) = 2\,436\,684\,974\,110\,751 \approx 2.4 \times 10^{15}$. Therefore, efficient algorithms for clustering are needed.

The clustering algorithms can be divided into a few types. Anderberg suggests the division into *hierarchical* and *non-hierarchical* methods, and Hastie, et al. in their recent work (Hastie et al. 2001) suggest three types: *combinatorial algorithms*, *mixture modeling*, and *mode seeking*. Combinatorial algorithms do not assume any underlying probability model. Mixture models suppose that there is a probability distribution from which data have been sampled i.i.d. Once the model has been fitted using *maximum likelihood* or Bayesian methods, the distribution can be characterized as a mixture of parametrized component densities. Mode seekers instead just attempt to directly locate the bumps in the probability density.

The goal of projection methods is to locate a linear or non-linear d_v-dimensional approximation for the d_x-dimensional data by minimizing the representation error. Typically $d_v \ll d_x$, which makes the representation easier to interpret than the original data. The data points are projected onto a line (*Principal Component Analysis*, PCA), curve (*principal curves* (Hastie and Stuetzle 1989)), plane, surface (*principal surfaces* (Hastie and Stuetzle 1989, Hastie et al. 2001)), or cluster centroids (several methods exist for this, for example the Self-Organizing Map (see below), *Generative Topographic Mapping* (GTM) (Bishop et al. 1998), *kernel-based topographic map* (Van Hulle 2001), and so on).

The Self-Organizing Map (SOM) (Kohonen 1982, Kohonen 1997) is a method,

which combines clustering and projection. It divides the data into clusters, which belong to a typically 1-D string or 2-D lattice, which can be used to visualize the contents of and relations between the clusters. This property makes the SOM an efficient visualization tool. Figure 3 illustrates the differences between a few commonly used clustering and projection methods, k-means clustering, Principal Component Analysis, principal curve, and the Self-Organizing Map.

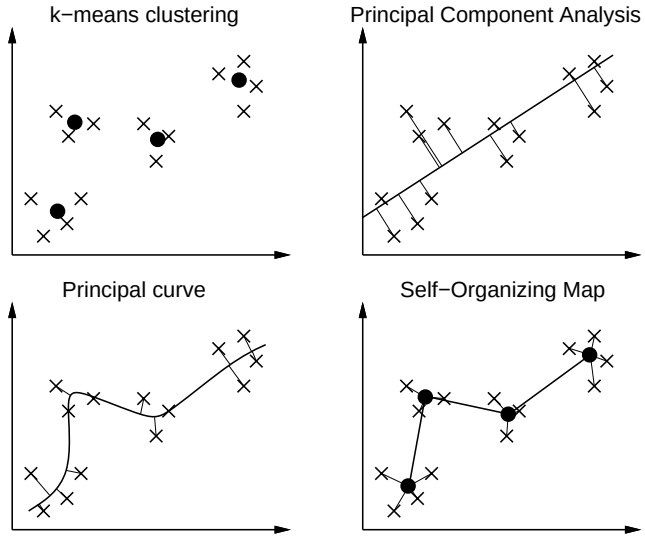

FIGURE 3: Some frequently used clustering and projection methods applied to 2-D data.

2.4 The Self-Organizing Map

The Self-Organizing Map has been successfully applied to the analysis of many different data types by developing clever preprocessing and feature extraction methods, see (Kohonen 1997). The SOM representation of data can be characterized as a nonlinear manifold, or as a principal surface (Hastie et al. 2001), whose dimension, d_v, is usually one or two, and thus lower than the dimension, d_x, of original data. The points of this surface, $\boldsymbol{v} \in \mathbb{R}^{d_v}$, go through the mean of the data $\boldsymbol{x} \in \mathbb{R}^{d_x}$ such that $\boldsymbol{x}(\boldsymbol{v}) = \mathbb{E}[\boldsymbol{X} \mid \boldsymbol{v}'(\boldsymbol{X}) = \boldsymbol{v}]$.

A rigid surface cannot go through all data points, and therefore the model has a representation error $\boldsymbol{\varepsilon} = \boldsymbol{X} - \boldsymbol{x}(\boldsymbol{v})$, which can be interpreted as noise or quantization error. A very flexible surface would be able to represent all data points directly, but then the surface could become folded and the complexity of the representation high, which would make the interpretation of the model difficult. In the Self-Organizing Map this principal surface $\boldsymbol{x}(\boldsymbol{v})$ is approximated with a finite rectangular or hexagonal lattice of nodes, *neurons*, $\hat{\boldsymbol{v}}(k), k = 1, 2, \ldots, N_v$, which act as cluster prototypes. Figure 4 illustrates a two-dimensional SOM,

which has been trained using 3-D data. The same figure also illustrates how the SOM can be visualized.

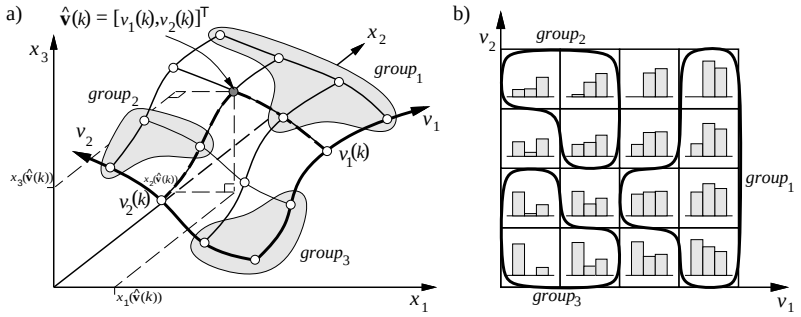

FIGURE 4: A 2-D SOM trained using 3-D data. a) In the data space the SOM attempts to follow data distribution, but still keeps the neighboring neurons similar to each other. b) The SOM surface can be used for the visualization of training and background data. The bars depict the coordinates x_1, x_2, and x_3 of the neurons.

A SOM with a large number of neurons typically corresponds to a flexible principal surface while a small number of neurons makes the surface rigid. When using the SOM the exact number of distinct clusters in the data need not be known, because a large amount of similarity between several cluster prototypes indicates that a smaller number of clusters might suffice, and intra-cluster dissimilarity suggests that a larger SOM should be used. Therefore, the number of neurons for a particular problem has typically been chosen intuitively, and the result has been evaluated using visualization of intra-cluster similarity measures. There are also quantitative criteria (see Chapters 3.2 and 3.3) with which the choosing can be performed without human intervention.

Each neuron $\hat{\boldsymbol{v}}(k)$ is associated with a prototype (or *reference* or *weight*) vector $\boldsymbol{w}(k) = [w_1(k), w_2(k), \ldots, w_{d_x}(k)]^{\mathrm{T}}$, whose d_x weight values correspond to the elements of the data vectors $\boldsymbol{x}(j) = [x_1(j), x_2(j), \ldots, x_{d_x}(j)]^{\mathrm{T}}$. The purpose of the *training* process is to iteratively find positions $\boldsymbol{w}(k)$ for the neurons $\hat{\boldsymbol{v}}(k)$ in the data space $\mathbb{R}^{d_x}$ such that the SOM *potential function*, E, is minimized. The potential function is calculated using

$$E = \sum_j \sum_k h_{c,k} \|\boldsymbol{x}(j) - \boldsymbol{w}(k)\|^2 \text{ , where} \quad (2)$$

$$c = c(\boldsymbol{x}(j)) = \arg\min_k \|\boldsymbol{x}(j) - \boldsymbol{w}(k)\|^2 \, . \quad (3)$$

There is a shrinking *neighborhood kernel*, $h_{c,k}$, in this function, and it is used also for the updating of $\boldsymbol{w}(k)$ in order to make the lattice smooth in data space. The distance measure is typically Euclidean, but other measures (Manhattan, Hamming, etc.) could also be used.

2.4.1 Effective Self-Organizing Map Training

One effective variation of the SOM is a tree-structured training algorithm, the TS-SOM (Koikkalainen and Oja 1990, Koikkalainen 1995, Koikkalainen 1999). In this batch algorithm several SOMs are built starting from simple SOM models and advancing to more complex ones. Each SOM model is called a TS-SOM *layer*, indexed by ℓ, and where the number of neurons $N_v = 2^{d_v \ell}$. Each layer is fixed after it has been trained. Using a tree search through the simpler layers, the correct area of the current layer can be found without performing a full search of all neurons. Therefore, the TS-SOM improves the training in its most critical step and reduces the computational complexity of the search from $\mathcal{O}(d_x \times N_x \times N_v)$ to $\mathcal{O}(d_x \times N_x \times \log N_v)$.

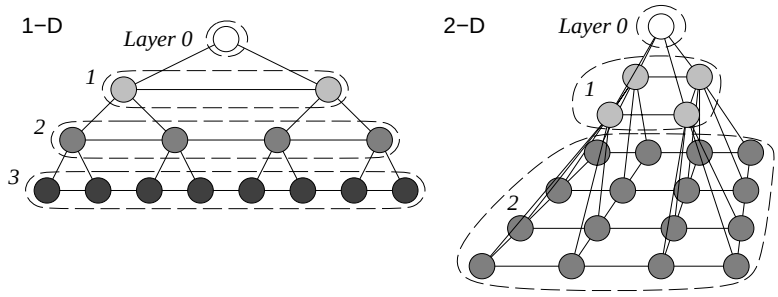

FIGURE 5: 1-D and 2-D TS-SOM structures.

Figure 5 illustrates the 1-D and 2-D TS-SOM structures. Each layer is either a 1-D string or a 2-D lattice of neurons. Since the models are simple in the beginning and eventually become more complex, the neighborhood kernel can be fixed to only the two (1-D TS-SOM) or four (2-D TS-SOM) closest neighbors, because the previously trained models, and previous layer based initialization of neuron weights ensure topological ordering of the lattice. The hierarchical structure also reduces the need to adjust or optimize the SOM training parameters (training speed and neighborhood size).

A significant advantage of the TS-SOM is the availability of several model resolutions. All these models can be later used for the visualization of training or background data. Using a simple layer, an overview of the contents of the data can in many cases be obtained, and the more complex models are able to represent small clusters, such as outliers.

2.5 Fuzzy Set Theory

Fuzzy set theory (Zadeh 1965, Zimmermann 1985) was proposed in order to make the definition of uncertainty possible, while working with sets. Traditionally, sets have a crisp definition where an object either belongs to a set or it does not. In the Zadeh's approach an object can belong to a fuzzy set with a degree (or grade)

of *membership* between 0 (does not belong) and 1 (belongs). Formally, fuzzy sets are defined as a set of ordered pairs $\tilde{s} = \{(x, \mu_{\tilde{s}}(x)) \,|\, x \in \Omega\}$, where Ω denotes a collection of items (typically real or discrete numbers), and $\mu_{\tilde{s}}(x)$ is the fuzzy membership function, which maps the items x of Ω to a membership space, which is typically the range $[0, 1]$. The membership function $\mu_{\tilde{s}}(x)$ can be defined as a formula, such as

$$\mu_{\tilde{s}_1}(x) = \frac{1}{1 + x^2} \tag{4}$$

for "Real numbers close to 0". A discrete definition is also possible, such as for $x \in \{1, 2, 3, 4\}$, the fuzzy set "Discrete value close to 3" could be specified with

$$\tilde{s}_2 = \{(1, 0.25), (2, 0.5), (3, 1.0), (4, 0.5)\} \; . \tag{5}$$

Figure 6 illustrates these fuzzy sets.

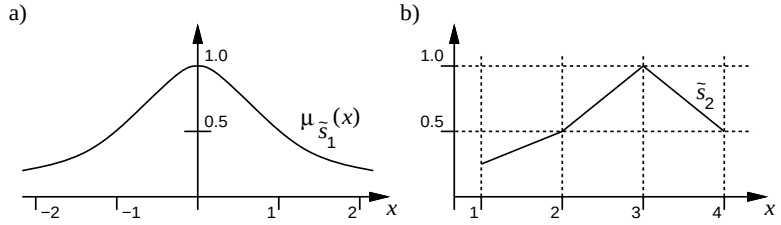

FIGURE 6: The example fuzzy sets. a) "Real numbers close to 0". b) "Discrete value close to 3".

Another advantage of the use of fuzzy sets is that linguistic interpretations can be attached to quantitative data. For example, by defining suitable fuzzy membership functions $\mu_{\tilde{s}_i} : x \rightarrow [0, 1]$ for three fuzzy sets $\{small, medium, large\}$ one can convert the value of x into a linguistic definition. If the memberships $\mu_{\tilde{s}_i}(x)$ were 0.7, 0.3 and 0.0 to the above mentioned sets, one could say that "the value of x is between $small$ and $medium$; closer to $small$".

Fuzzy membership functions can be overlapping, and there are no strict rules that they should sum up to 1.0 at each value of x. However, some mathematical formalisms for fuzzy sets include more strict requirements. Typical shapes for fuzzy membership functions are triangular, trapezoidal and Gaussian. If the Gaussian membership functions are used, and a probabilistic interpretation is needed, the functions need to fulfill $\int \mu_{\tilde{s}}(x)\, dx = 1$.

2.6 Statistical Natural Language Processing

A comprehensive review of *Natural Language Processing* (NLP) methods can be found in (Jurafsky and Martin 2000), which also covers speech recognition and text-to-speech techniques. A statistical approach to NLP is described in (Manning and Schütze 1999).

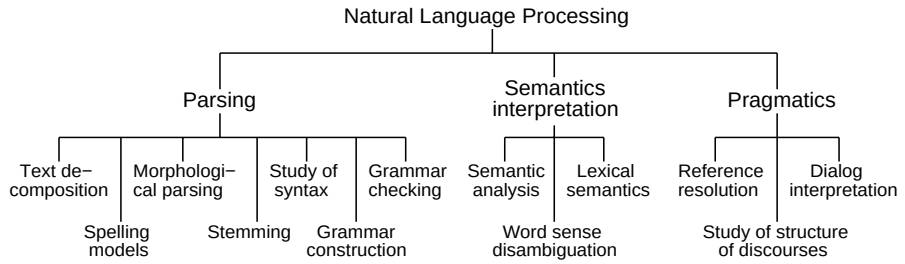

FIGURE 7: The taxonomy of natural language processing. The three main research areas are divided to numerous tasks, which are usually combined when building language processing systems.

NLP covers a wide range of research areas, including *parsing*, *semantics interpretation*, and *pragmatics* (Jurafsky and Martin 2000). The tasks related to each of these areas are illustrated in Figure 7. Almost always the text has to be decomposed into words, but models for spelling, morphological parsing (detection of inflectional or derivational forms), stemming (removal of morphological affixes), study of syntax (word class detection or part-of-speech tagging), grammar construction, and grammar checking (or parsing with a grammar) are not necessarily needed in parsing. Semantics is the study of the meaning of the text, including disambiguation of word sense, semantic analysis (the building of meaning representations and assigning them to textual data), and lexical semantics (detection of synonyms, homonyms, hyponyms, hypernyms, antonyms, etc.). Finally, pragmatics is '*the study of how knowledge about the world and language conventions interact with literal meaning*' (Manning and Schütze 1999). For example, the study of the structure of discourses, reference resolution, and dialog interpretation are tasks in the area of pragmatics.

In statistical NLP the main goal is to describe the linguistic content using probabilistic models. The models can be used for many tasks in parsing and semantics interpretation, including word sense disambiguation and part-of-speech tagging. Due to the availability of large masses of textual data in electronic form, three key areas have emerged to assist the locating of relevant information: *information retrieval*, *text categorization*, and *text clustering* (or *text mining*).

The complexity of a natural language is usually illustrated by calculating the cross entropy of the probability distributions indicating the belief, what the next character, c_r, is when κ previous characters are known, $\Pr(c_r \mid c_{r-1}, c_{r-2}, \ldots, c_{r-\kappa})$ (Shannon 1948). Shannon used an alphabet of 26 letters and the *space* for the English language (case distinctions and punctuation were omitted). This kind of probability model can also be used for generating text, whose quality typically increases when κ gets larger if enough model training material is available.

Similar models can also be built for the words, ω_i, of the documents $\Pr(\omega_n \mid \omega_1, \omega_2, \ldots, \omega_{n-1})$. These models are called word n-gram models (*bigram*, *trigram*, and so on) according to the number of previous words that are taken

into account. These models become very complex, and a lot of data is needed for their building, if the vocabulary is large. Therefore, the n-gram model typically is not the best approach for similarity detection from textual data, especially if the word order is not strict.

2.6.1 Modern Information Retrieval

Information Retrieval (IR) (Baeza-Yates and Ribeiro-Neto 1999) differs from data retrieval (from databases for example), because the main motivation is not to just satisfy a given query, but instead retrieve *relevant* information about a subject. So, even though queries are frequently used in IR to specify what the user is interested in, the goal in IR is to organize or *rank* the potential results according to expected relevance. The classical IR models are called *Boolean*, *vector*, and *probabilistic*.

All these models are based on a set of index *terms* (or *keywords*), $\mathcal{T} = \{t_1, t_2, \ldots, t_{N_t}\}$, whose semantics is important for the contents of those documents in which they occur. Usually the order of words is not taken into account, and thus the text documents, d_j, are described using *index term vectors*, $\boldsymbol{d}(j) = [w_1(j), w_2(j), \ldots, w_{N_t}(j)]^{\mathrm{T}}$, which indicate with weights $w_i(j)$ the relevance of all terms $t_i \in \mathcal{T}$, for d_j. There are many heuristic methods for choosing the set of index terms. It is at least quite usual to remove the most common *stopwords* (such as prepositions, particles, and so on), and in many cases also those regular words, which occur in almost all documents (because they do not improve retrieval performance).

Another way to reduce the size of the index term set is to use *Latent Semantic Indexing* (LSI) (Deerwester et al. 1990). In this method, a document vs. index term matrix is decomposed into parts by using *Singular Value Decomposition* (SVD), and only the most important informative terms are retained. The calculation of frequencies can be a difficult problem in languages with rich morphology due to the large number of different forms of the same word. However, *stemming* algorithms can be used in converting the words into their basic form.

In the Boolean models document descriptions, $\boldsymbol{d}(j)$, are binary vectors, which means that either an index term occurs in the document, 1, or not, 0. Retrieval is based on expressions, which are specified using Boolean logic ($\wedge$, $\vee$, $\neg$), and which can be converted into the *disjunctive normal form*, which specifies whether a subset of $\mathcal{T}$ should occur in the document or not. Boolean models are simple and the formalism is clean. However, they lack the ability to make partial matches, and thus cannot rank the results.

The vector model (Salton and McGill 1983) uses non-binary weights, $w_i(j)$, for the index terms both in the query, $\boldsymbol{q}$, and in the document description, $\boldsymbol{d}(j)$, vectors. The query vector consists of N_t weight values, w_i^{query}, and *similarity*, $sim(d_j, query)$, is typically assessed by calculating the *cosine of the angle* between

the query and document vectors

$$sim(d_j, query) = \frac{\boldsymbol{d}(j) \cdot \boldsymbol{q}}{\|\boldsymbol{d}(j)\| \, \|\boldsymbol{q}\|} \, , \tag{6}$$

where the numerator is a dot product, and the denominator multiplies the vector lengths. The value of similarity varies from 0 to 1, and allows partial matches and the ranking of documents.

The index term weight, $w_i(j)$, for each document, d_j, are usually calculated by multiplying the relative *term frequency* $tf_i(j)$ within the document by *inverse document frequency* idf_i within the corpus

$$w_i(j) = tf_i(j) \times idf_i = \frac{freq_i(j)}{\max_l freq_l(j)} \times \log \frac{N_d}{n_i} \, , \tag{7}$$

where $freq_i(j)$ indicates how many times t_i occurs in d_j, N_d is the total number of documents, and n_i indicates the number of documents in which t_i appears. Several other weighting methods exist, but *tf-idf* based schemes have been the most popular.

The probabilistic models use relevance information from the user for refining the set of retrieved documents. Also other *term reweighting* schemes exist, with which retrieval performance can be improved. The number of retrieved relevant documents can also be increased by query expansion techniques, where the assumption is made that relevant documents contain frequently the same terms, and non-relevant documents have different kind of index term vectors. Suppose that the set of documents, which match some query, q, have also some other common terms, t_i, which are not included in q, but whose weights $w_i(j)$ are high. These terms could be added to the query, and it could be re-evaluated to find also such relevant documents, which do not match the original query.

The Information Retrieval models are typically evaluated using two measures: *precision* and *recall*. Precision indicates the fraction of the retrieved documents, which are relevant to the query

$$precision = \frac{\#(relevant\_retrieved)}{\#(retrieved\_documents)} \, . \tag{8}$$

Recall indicates the fraction of all relevant documents retrieved

$$recall = \frac{\#(relevant\_retrieved)}{\#(relevant\_documents)} \, . \tag{9}$$

Figure 8 illustrates the sets required for calculation. Both of these measures require that the judgment of relevance is assessed for the retrieved documents, and $recall$ also requires that the total number of relevant documents must be known.

2.6.2 Classification and Clustering of Text Documents

In text categorization (Sebastiani 2002) the task is to classify text documents into predefined categories. Rule based and supervised statistical learning methods

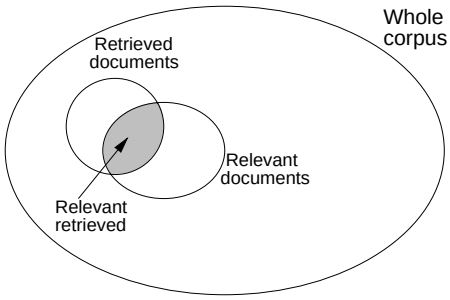

FIGURE 8: The sets used for the calculation of *precision* and *recall*.

have been suggested for the classification task, such as probabilistic, decision rule based, decision trees, neural networks (Bishop 1995), Support Vector Machines (Vapnik 1995), and committee models. However, because a category structure is in many cases not available, and cannot be easily constructed, unsupervised statistical learning techniques have also been applied. This kind of approach where similarities are detected from text documents without a predefined classification is called text clustering or text mining. Text categorization methods are especially useful for the filtering or classification of incoming information, and text clustering methods for browsing or exploring large collections of unfamiliar content.

In text categorization and text clustering, many approaches use index term based description techniques as in IR, but quasi-orthogonal random vectors and word context based (Ritter and Kohonen 1989, Kaski et al. 1996) methods have also been suggested. Text categorization has the advantage that the document classification information can be used for the choosing of the set of most informative or separative terms, $\mathcal{T}^{inf}$. This kind of optimization may improve the categorization of the training data set a lot, but may have negative effects for the testing. In text clustering such exogenous information is not available, and the term set has to be chosen otherwise.

Text categorization methods are typically evaluated using the same measures as in IR methods, $precision$ and $recall$, but the interpretations are somewhat different. If the system is asked to retrieve all documents which belong to certain category, $precision$ indicates the fraction of retrieved documents which are correctly classified to the chosen category, and $recall$ expresses the fraction of the whole category retrieved. Text clustering methods cannot be evaluated the same way if no background knowledge about the documents exists.

For the analysis of completely new data, unsupervised text clustering methods are applicable. A Self-Organizing Map based model has the advantage to other clustering methods in that the clusters are ordered according to similarity, and therefore the map can be used for searching similar content. A query can first be used to locate a good starting point, and then map units close to that point can be explored to find documents which are similar in content. This could be seen as a query expansion technique, because such documents can be found, which

do not match the query, but which still have other common features with those documents matching the query.

SOM exploration of text documents have been proposed by several authors. Quasi-orthogonal random vectors were suggested in (Ritter and Kohonen 1989), and have been applied by Kohonen et al. (Honkela et al. 1995, Kohonen et al. 1996, Kaski et al. 1996, Lagus et al. 1996, Kohonen 1997, Lagus 1997, Honkela et al. 1997, Honkela 1997, Kaski et al. 1998, Kohonen 1998, Lagus 1998, Kohonen et al. 1999, Lagus et al. 1999, Kohonen et al. 2000). With some corpora also contextual information has been used for document description. This method has been called the WEBSOM since 1996. Miikkulainen has developed a system called DISCERN (Miikkulainen 1993, Miikkulainen 2000), which uses the *Hierarchical Feature Map* (Miikkulainen 1990) for similarity detection. Merkl has experimented with MLP neural network based compression of input vectors (Merkl 1995), the use of the Hierarchical Feature Map (Merkl 1997a, Merkl 1997b), and *tf-idf* weighted terms (Rauber and Merkl 1998, Rauber and Merkl 1999, Merkl 1999). Bernard introduced a method for detecting *word classes* from text written in French language (Bernard 1997). And finally, Kurimo and Lagus (Kurimo and Lagus 2002) suggested an improved word co-occurrence model to combine local n-gram statistical models.

The size of the training corpora has varied from a few hundred documents up to almost 7 million with average lengths ranging from a few to a few hundred words. The chosen preprocessing method has typically been chosen after the statistical properties of the corpus have been assessed. In many cases, stopwords are removed manually or using a stopword list, and some of the most frequent and most rare words have been removed from the index term set.

2.7 Visualization of Data and Models

> There is no more reason to expect one graph to "tell all"
> than to expect one number to do the same.
>
> *John Tukey*

Several advantages and limitations of graphs are listed in (Tukey 1977) p. 157 for plots of y against x, such as '*Graphs force us to note the unexpected; nothing could be more important*', and the quotation above. It can be even more difficult to understand graphs when data dimensionality is higher. But by using clustering and projection methods for dimensionality reduction, the visualizations can be made more understandable. However, when visualizing one should keep in mind that visual presentations must be easy to read, and their contents must be explained thoroughly. This is essential, because in many cases the final conclusions are drawn by the person who has collected the data and who has the necessary background knowledge, but who is not a statistician nor an expert in computer science.

Tukey lists two main motivations for exploratory analysis:

- *'Anything that makes a simpler description possible makes the description more easily handleable.'*
- *'Anything that looks below the previously described surface makes the description more effective.'*

He further continues that even *'to be able to say that we looked one layer deeper, and found nothing, is a definite step forward – though not as far as to be able to say that we looked deeper and found thus-and-such'*. All of these motivations support the use of a system with which it is possible to interactively browse graphs, whose contents can be chosen and which allow the user to choose parts of the data for further studies.

Miles and Huberman have an even stronger opinion about visualizations in the context of qualitative data analysis (Miles and Huberman 1994). They say that *'The dictum "You are what you eat" might be transposed to "You know what you display." '*, which stresses the advantages of their descriptive approach. Miles and Huberman also motivate using an interactive approach to data analysis using the diagram in Figure 9. This diagram includes the data collection phase, because in their methodology reduction and display are used already while collecting the data, in order to be able to decide if the collecting should be simplified. For already collected data this model can be used without data collection. What remains is a completely interactive model.

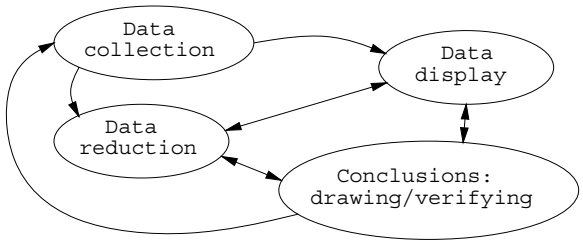

FIGURE 9: The interactive model of data analysis adopted from (Miles and Huberman 1994).

2.7.1 Visualization of the SOM

Since the Self-Organizing Map divides the training data into clusters, which are ordered according to a (typically) two-dimensional lattice, this lattice can be used to efficiently describe the statistical properties of the clusters. Several basic and advanced SOM visualization techniques have been suggested for numerical data, see for example (Kohonen 1997, Häkkinen and Koikkalainen 1997a, Häkkinen and Koikkalainen 1997b, Vesanto 1999, Häkkinen 2001).

The lattice can be visualized as a two-dimensional graph, where each cluster is represented by a square, a circle, or a hexagon (if the hexagonal lattice is used),

within which the statistical information can be rendered. For each cluster one can, for example, calculate the averages and variances, or locate the minimums, quartiles, medians, and maximums of the training and/or background variables. Then these values can be visualized as background color (one variable at a time), as surface height in 3-D (one variable), or as line or bar graphs (several variables at the same time). The use of a distance matrix for varying the neuron rectangle sizes can be useful for indicating large distances between neighboring neurons. Another method for showing the (approximate) distances between neurons (or cluster centroids) is to use *Sammon's mapping* (Sammon Jr. 1969) or two principal components to project the multivariate cluster prototype vectors, $\boldsymbol{w}(k)$, into a two-dimensional space. However, this kind of illustration is not able to show all distances accurately if the original data has more than two dimensions and the variables are not correlated. Illustrations of possible SOM visualizations can be found from the articles, especially [H] and [I].

In addition to fancy graphics, the interactivity of the user interface is also one of the key features of an exploratory analysis system. Interactive tools may allow the user to

- Select a neuron in order to get to see the raw data classified to that neuron.
- Compose groups of neurons in order to obtain statistical information about all the data points classified to those neurons.
- Select data (using neuron prototypes) for more thorough analyses.
- Visually disclose data, which is similar to a query.

2.7.2 Visualization of Text Document Maps

Text document maps are typically visualized using keywords, and document lists, which are opened according to user selections. Lagus and Kaski proposed a keyword selection method (Lagus and Kaski 1999), which is based on relative frequencies of a word, ω, within a cluster of the SOM, Ω_k. The goodness of each word is evaluated by noting the importance of ω for the current cluster k, and by finding out if the word is used more in this cluster than elsewhere. In order to take into account the fact that those units of the SOM lattice, which are close to k may also contain ω frequently, they also suggested another measure of goodness, which does not take into account the close clusters on the SOM lattice (the distance limit is chosen experimentally). The paper also contains a method for assigning representative words for larger map areas of several clusters using another measure.

Merkl and Rauber have also proposed an automatic labeling method for SOM based document maps (Merkl and Rauber 1999). This LabelSOM method is based on the simple fact that all those terms are listed, which are strongly represented in each cluster, Ω_k, and also in the neuron weight vector, $\boldsymbol{w}(k)$. Two criteria are needed in their approach, because the document description vectors are long, and the weight values may not quite directly represent the group of doc-

uments, because of the neighborhood smoothing used in the SOM. The keyword lists obtained with this method are longer than in the method proposed by Kaski and Lagus, and therefore cannot be shown as labels in a SOM visualization.

2.8 Discussion

Qualitative research is clearly different from all other presented methods due to the subjective human approach. Most other methods attempt to provide means for quantitative or visual study of the contents of data.

3 MODEL SELECTION

When the proposed methodology for the SOM based survey data analysis was introduced, many important problems were omitted. Perhaps the most important of these is the question about the complexity of the model versus real-world observations. In Article [H] we have illustrated that the use of TS-SOM allows one to model data with several resolutions. Simple SOMs are models that strongly average the data, whereas complex models are able to capture individual observations, including noise. Sometimes this is not a problem as it allows the human observer to capture the information from the data. But in automated analysis, for example in a text clustering task, it is most important that the complexity of the word category map is selected properly. This problem is studied in Article [G].

Formally the questions are:

- How to define the complexity of the SOM?
- How to select the complexity of the SOM for certain data?
- What are the effects of the complexity selection?

The first part of this chapter introduces general model selection and assessment methods, which are important while working with statistical learning methods. The next part introduces methods, which have been used for the evaluation of SOM models. And the last part describes our complexity selection method for the TS-SOM.

3.1 Model Selection and Assessment

> Pluralitas non est pondena sine necessitate.
>
> *William of Ockham*

In statistics a *true* distribution, which describes the characteristics of the data generating process, is often assumed, but the type and the number of parameters of the real-world processes are unknown. Therefore, statisticians have created many model evaluation methods, with which the plausibility or the prediction performance of models can be assessed. One should keep in mind that there are two aspects in this process, *model selection* and *model assessment* (Hastie et al. 2001). Model selection is the process of estimating the performance of models for given data, and assessment refers to the evaluation of the applicability of the model for other data sampled from the same real-world distribution.

The representation or prediction performance of a model depends on many things, for example model type, the number of parameters, and parameter values. The optimal parameter values are typically estimated (or *learned*) for each model before the evaluation. The quotation above (in English: '*Plurality should not be assumed without necessity.*', which could be interpreted as '*Keep it simple, stupid.*')

refers to one of the main applications of model selection methods, which is the choosing of the number of model parameters, when the model type has already been chosen. The basic idea of simplicity has also been frequently referred to as *Occam's razor* due to an idea of *shaving away all that is unnecessary*. This idea is called the *principle of parsimony* in statistics, and has traditionally been addressed by using the *bias-variance decomposition* of the model prediction error, and by locating a tradeoff between them.

The bias (prediction error at certain point x_0) of a very simple model is large and the variance (variability of the model) is small. If the number of parameters is gradually increased, the model is usually able to represent the data better all the time, and the bias gets smaller. However, at the same time the variability of the model typically increases. After certain number of parameters is reached, the *generalization performance* for new data begins to get worse (Hastie et al. 2001), see Figure 10. A solution to this problem is to divide the data into three separate sets: training, *validation*, and *test*. The models are fit using the training set, the "best" model is chosen by calculating the prediction error using the validation set, and the generalization error is estimated using the test set. However, there are situations, where either there is not enough data for the division or the prediction error cannot be calculated (if we are not predicting or classifying anything). If the amount of data is somewhat small, resampling techniques such as *cross-validation* (Stone 1974) or *bootstrap* (Efron 1979) can be used.

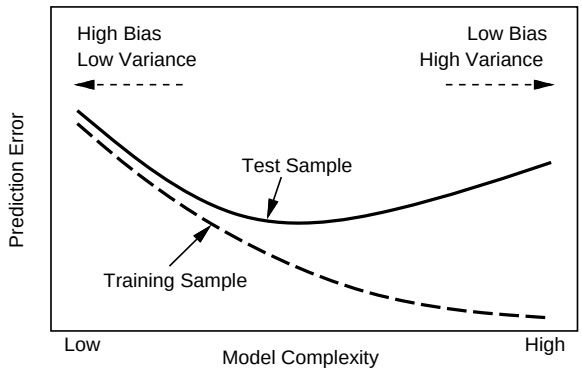

FIGURE 10: Behavior of test sample and training sample error as the model complexity is varied, adopted from (Hastie et al. 2001) p. 194.

The selection of the number of model parameters is essential for *universal approximators* (models, which can approximate any continuous functions) to avoid *over-fitting* the model to the specific training data. Also, the interpretation of the model is easier if the model is not too complex. These non-parametric models can be adapted to almost any data by increasing the number of parameters. The presented bias-variance decomposition, and the data division or resampling approaches can be used with many universal approximator models, too. However, the information theoretic model selection methods presented in this chapter have

the advantage that they can be used also in situations where the prediction error is not applicable.

3.1.1 Information Theoretic Methods

Information theory has traditionally been a part of communication theory, and was developed to give answers to two questions of utmost importance: *what is the ultimate data compression*, and *what is the ultimate transmission rate of communication* (Cover and Thomas 1991). These were already solved in (Shannon 1948) with the introduction of *entropy* and *channel capacity*. Entropy is a measure of uncertainty of a random variable, and channel capacity sets a limit for the communication rate. For a discrete random variable X with probability density function $p(x) = \Pr(X = x)$, where x belongs to the set, $\mathcal{X}$, of possible values of X, the entropy H (in bits) can be calculated with

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x) \ . \tag{10}$$

Shannon also showed that a sequence of symbols, where certain symbols occur more frequently than others, can be transmitted the most efficiently, if the frequent symbols are represented by short description strings, and rare symbols are given longer descriptions. Entropy actually gives the lowest achievable average number of bits per symbol needed to communicate some sequence of symbols to a receiver if their probability density is $p(x)$. Since the entropy represents a $p(x)$-weighted average of values $-\log_2 p(x)$, these (if they are integers) represent the optimal code lengths to be used for the symbols. The fact that codes of such lengths are truly optimal is proved in (Cover and Thomas 1991), where it is also shown that calculations can be done using fractions of bits, because the expected code length per symbol can be achieved using large blocks of symbols.

Statistical inference is a key application area of information theory, and several model selection methods have been developed. Information theoretic model selection methods have their roots in independent development of algorithmic probability by Solomonoff (in 1964) and algorithmic complexity by Kolmogorov (in 1965) and Chaitin (Chaitin 1966), see the discussion in (Kolmogorov 1968). The *complexity* of data, Ω, is defined as the length (in bits) of the minimal computer program (for a universal computer), which can generate Ω. Since the shortest computer program cannot be determined in practice, exact Kolmogorov complexity cannot be calculated. Practical implementations of model selection use another approach where some class of models is used for data description instead of the universal computer. The simplicity and sufficiency of the model are evaluated at the same time either by comparing the data distribution to the model or by constructing a decodable message for each candidate model and *residuals* ($observed\ data = model\ fit + residuals$), which indicate how much the actual data, Ω, differ from the model. If the model is suitable, then the data can be described

using the model with substantially less bits than would be needed for the description of the original data.

3.1.2 AIC and BIC

Akaike's Information Criterion (AIC) by Akaike (1973) and *Bayesian Information Criterion* (BIC) (Schwarz 1978) are criteria with which models from different families can be compared to the distribution of data to be fitted. A good description of AIC can be found from (Burnham and Anderson 1998). AIC is based on *Kullback-Leibler distance* (or *discrepancy* or *divergence*) (Kullback and Leibler 1951), which is a directed distance between the true model f and some candidate model g with parameters $\boldsymbol{\theta}$, which can be optimized for the data using least squares or maximum likelihood methods to obtain estimated parameters $\hat{\boldsymbol{\theta}}$. The Kullback-Leibler (or K-L) distance is defined as

$$I_{KL}(f, g) = \int f(y) \log \left(\frac{f(y)}{g(y \mid \boldsymbol{\theta})} \right) dy \ , \tag{11}$$

where y is just an integration parameter.

The K-L distance can be interpreted as the amount of "information", which would be lost, if g is used for approximating f (Burnham and Anderson 1998), or *'inefficiency of assuming that the distribution is g when the true distribution is f'* (Cover and Thomas 1991). The K-L distance $I_{KL}(f, g) \geq 0$ always, and it is 0 iff $f(y) = g(y) \ \forall \ y$. Figure 11 illustrates two candidate distributions g_1 and g_2, and a true distribution f.

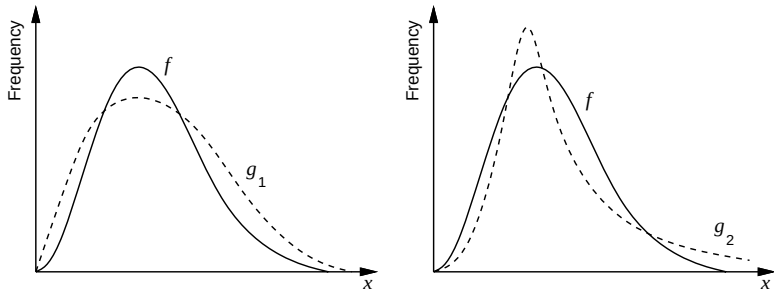

FIGURE 11: An example of comparing f and g_i. The idea to this figure is adopted from (Burnham and Anderson 1998) p. 39.

If the true distribution f, from which data $x \in \Omega$ are sampled, was known, K-L distance could be directly used to compare f to the distributions obtained using some candidate models $g_i(x \mid \hat{\boldsymbol{\theta}})$ with optimized parameters. In data analysis the true (real-world) distribution is unknown (only data is available), and thus the exact K-L distance cannot be calculated. However, Akaike noticed, that the

expected estimated K-L distance

$$\mathbb{E}_{\hat{\boldsymbol{\theta}}}[\hat{I}_{KL}(f,g)] = \int f(x) \left[\int f(y) \log \left(\frac{f(y)}{g(y \mid \hat{\boldsymbol{\theta}}(x))} \right) dy \right] dx \ , \tag{12}$$

where $\hat{\boldsymbol{\theta}}(x)$ are the optimized parameters for data x, could be written into another form

$$\begin{aligned} \mathbb{E}_{\hat{\boldsymbol{\theta}}}[\hat{I}_{KL}(f,g)] \ &= \ \int f(x) \left[\int f(y) \log(f(y)) \, dy \right] dx \\ &\quad - \int f(x) \left[\int f(y) \log(g(y \mid \hat{\boldsymbol{\theta}}(x))) \, dy \right] dx \ , \end{aligned} \tag{13}$$

which can be used for relative K-L distance calculation, since the first part is constant (f and data do not change). The second part can also be written as the expectation $\mathbb{E}_x \mathbb{E}_y[\log(g(y \mid \hat{\boldsymbol{\theta}}(x)))]$, which can be estimated for large samples using a reduced log-likelihood, $\log(\mathcal{L}(\hat{\boldsymbol{\theta}} \mid x)) - K$, where K is the number of estimated parameters. For historical reasons, this estimation is multiplied by -2 to obtain *Akaike's Information Criterion*

$$AIC = -2 \log(\mathcal{L}(\hat{\boldsymbol{\theta}} \mid x)) + 2K \ . \tag{14}$$

The Bayesian Information Criterion

$$BIC = -2 \log(\mathcal{L}(\hat{\boldsymbol{\theta}} \mid x)) + (\log(N_x))K \ , \tag{15}$$

where N_x denotes the number of samples in the data, Ω, is very similar to AIC, but penalizes complex models more heavily (Hastie et al. 2001). Even though the criteria are similar, the motivations for BIC are completely different. BIC is based on a Bayesian approach where ratios of posterior probabilities are calculated.

The problem with these approaches is that they cannot be used with non-parametric models, unless strong assumptions are made. This is because otherwise the calculation of likelihood (Edwards 1972) or the estimation of the effective number of parameters are not possible.

3.1.3 Minimum Description Length

Minimum Description Length (MDL) (Rissanen 1987, Rissanen 1989) was suggested by Rissanen in 1978. In order to locate the best model from some proposed class, the code lengths of the data using models from the class are evaluated using a coding approach. *Stochastic complexity* is the shortest possible code length for the model class.

In the basic MDL a two-part coding is used, in which a model m_i with parameters $\boldsymbol{\theta} = \{\theta_1, \theta_2, \ldots, \theta_K\}$ is first described using a codeword $C(\boldsymbol{\theta})$ of length $L(\boldsymbol{\theta})$, and then description of data $\Omega = \{x(1), x(2), \ldots, x(N_x)\}$ using m_i is given

another codeword $C(\Omega \,|\, \boldsymbol{\theta})$, whose length is $L(\Omega \,|\, \boldsymbol{\theta})$. The full description is the concatenation of the codewords

$$C(\Omega, \boldsymbol{\theta}) = C(\boldsymbol{\theta})\, C(\Omega \,|\, \boldsymbol{\theta}) \;, \tag{16}$$

and thus the description length can be calculated as

$$L(\Omega, \boldsymbol{\theta}) = L(\boldsymbol{\theta}) + L(\Omega \,|\, \boldsymbol{\theta}) \;. \tag{17}$$

In Rissanen's coding there are some complications due to parameters, which are real numbers. Thus parameters θ_i have to be truncated to precisions δ_i, which can be optimized and should be transmitted with the model. For large data sets ($N_x \gg K$) their impact is negligible, and thus for parametric probabilistic models the Minimum Description Length criterion can be reduced into a familiar form

$$MDL = -\log(\mathcal{L}(\hat{\boldsymbol{\theta}} \,|\, x)) + \frac{K}{2}(\log(N_x)) \;, \tag{18}$$

which, if multiplied by 2, is the same as BIC.

However, here the same restriction of parametric models applies as in AIC and BIC. Therefore, Rissanen has also suggested two two-part coding approaches for non-parametric model classes. The simpler approach is based on parametric histogram densities

$$f_H(x \,|\, p) = \frac{N_m p_i}{R} \;, \tag{19}$$

where N_m is the number of pins, R is the length of the interval in the real line within which the data fall, p_i are the parameters ($\sum_i p_i = 1$), and i is the index of the pin containing x. Figure 12 illustrates this kind of histogram density.

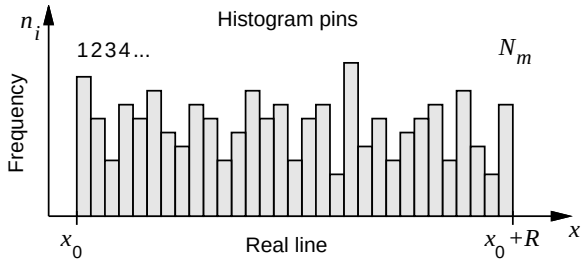

FIGURE 12: Illustration of a histogram density on a real line.

For a histogram, where the number of data points, n_i, within each pin i is roughly equal, Rissanen derived the stochastic complexity

$$I_{SC}(\Omega \,|\, N_m, N_x, R) = \\ N_x \log \frac{R}{N_m} + \log \binom{N_x}{n_1, \ldots, n_{N_m}} + \log \binom{N_x + N_m + 1}{N_x} \;, \tag{20}$$

where the second term is a logarithm of a multinomial, and the last term is a logarithm of a binomial. However, Rissanen stresses that a better density estimator exists, if the pin frequencies, n_i, are not almost equal.

The other approach is based on Gaussian kernels $f_G(x \mid \mu_i, \sigma_i)$, which are summed to get the density

$$f_D(x \mid \boldsymbol{\mu}, \boldsymbol{\sigma}) = \frac{1}{N_m} \sum_{i=1}^{N_m} f_G(x \mid \mu_i, \sigma_i) \ , \tag{21}$$

where N_m is the number of kernels, μ_i are the N_m-quantile points of ordered observations $x(j)$, and σ_i are variances calculated from truncated μ_i using $\sigma_i = {}^1\!/_4\,(\tilde{\mu}_{i+1} - \tilde{\mu}_{i-1})$, where $\tilde{\mu}_i$ denotes the truncated parameter. This behaves better than the histogram approach with non-uniform densities, but is still problematic to calculate, if the data is highly peaked (because some σ_i may become 0).

3.1.4 Minimum Message Length

Minimum Message Length (MML) (Wallace and Boulton 1968) was originally suggested for comparing clustering results, and the idea was later developed in (Wallace and Freeman 1987). It has also been referred to as *minimum encoding inference* (Oliver and Hand 1994), and its similarities and differences to MDL have been studied (Baxter and Oliver 1994). The basic MML is rather easy to calculate, see (Oliver and Hand 1994), and it does not *require* that the data should have a predefined distribution.

The basic idea of using Minimum Message Length for statistical inference and especially model selection (Oliver and Hand 1994) is to find a model m_i from some set of models, $\mathcal{M} = \{m_1, m_2, \ldots\}$, which is able to represent some specific data set Ω with the smallest number of bits. To evaluate the models, a two-part coding scheme is used, and an encoded binary string, $C(\Omega, m_i)$, is constructed, which contains the descriptions of m_i and $\Omega \mid m_i$. In practice the coding of the model parameters and the data using the model may require several code strings, which are catenated together, for example see the clustering evaluation example in (Hansen and Yu 2001). The MML criteria suggested in (Wallace and Boulton 1968) and (Wallace and Freeman 1987) for parametric models are somewhat complicated, because each parameter is given a different *Accuracy of Parameter Value* (AOPV). Therefore, it is not easy to compare MML to the other methods.

Since the exact complexity of the description cannot in general be determined (the coding used is *not* the most optimal), MML uses a pragmatic approach in which the complexity is approximated by the shortest string obtained with the most efficient known method. The encoded result might not be the most minimal description for the data using the model m_i, but as long as the same encoding methods are used consistently for all models in $\mathcal{M}$, the most suitable model from

that set can be found. The problem with MML is that it cannot be used for choosing the model class the same way as AIC, BIC, or MDL, see (Rissanen 1989) pp. 56–57.

3.2 Evaluation of the SOM

Many people have suggested experimental and analytic methods to evaluate a trained Self-Organizing Map. However, most of these methods only concentrate in choosing the best training parameters or optimizing the topology of a growing grid SOM. There are two main parameters in regular SOM training algorithms, which affect the iteration in SOM training: training speed, $\alpha(t)$, and neighborhood size (including its shrinking speed), $\sigma(t)$, where t is the iteration number. Kohonen presents a Voronoi tesselation motivated "optimized" learning rate factor (Kohonen 1997), and for the neighborhood size he suggests that the initial radius should be close to or even more than half of the network width, and that it should shrink linearly down to only a single unit. A semi-empirical learning rate has also been suggested for the regular SOM (Cherkassky and Mulier 1998).

Heskes suggests a division of evaluation methods (Heskes 1999): quantifying topology preservation, convergence proofs, and energy function explanations for the learning rule. To evaluate how well a map preserves the topology of the input space, a geometrical approach was suggested in (Zrehen 1993), which uses Voronoi regions of the neurons. Kaski and Lagus suggest using the average distance from data points through the closest neuron along the map to the second closest neuron of the SOM, in order to find good training parameters (Kaski and Lagus 1996); Herrmann, et al. review several topology preservation measures in order to select optimal model dimensionality, d_v^{opt}, for given data (Herrmann et al. 1997); and Polani reviews several organization measures (Polani 1997). Topology preservation partially also relates to the *growing grid* type of SOMs, such as GSOM implementations suggested in (Villmann and Bauer 1997, Villmann 1999), where the possibility of having hypercubical structure is constrained to lower dimensional latent spaces depending on the properties of the training data. Der and Herrmann demonstrate how the training parameters can be optimized by learning (Der and Herrmann 1999), and Polani suggested a genetic algorithm based optimization of parameters using topology measures (Polani 1999). However, none of these approaches attempt to choose the number of neurons for a certain problem automatically.

There are two approaches where the number of neurons has been chosen according to the properties of the training data. Hyötyniemi proposed a method based on Minimum Description Length (see 3.1.3) by assuming that the SOM can be interpreted as a mixture model of Gaussian distributions (Hyötyniemi 1997). In his paper there was an example, where the number of neurons was automatically selected in a situation where there were four Gaussian distributions. The applicability to real-world problems was not quite apparent. Cottrell, et al. pro-

posed the use of *coefficient of variation*, $CV(\theta) = 100\,{}^{\sigma_\theta}/_{\mu_\theta}$, where σ_θ denotes the variance and μ_θ the mean of θ, calculated from *intra-class sum of squares* of SOMs created by resampling the training data (Cottrell et al. 2001). They used the measure both for the evaluation of clustering stability, and choosing the "right" number of units for a SOM. However, their example is also simple, and the results are not easily interpretable.

3.3 Selecting Number of Neurons for TS-SOM

Since many data sets do not come directly from nature, the assumptions made by parametric modeling approaches may contradict with the properties of the multimodal high-dimensional data, which is to be analyzed or explored. However, if the idea of evaluating message length is applied in a non-parametric way to compare clustering results, the most suitable TS-SOM layer for given data can be chosen as illustrated in Article [G]. The same kind of approach could also be used for the regular SOM, but then the optimization of training parameters, selection of model sizes, and so on, would make the process difficult.

According to Baxter and Oliver, the main difference between MDL and MML approaches is that MDL is used to select both the model class (for example univariate quadratic vs. univariate cubic) and the best model from it, while MML just picks one model from the specified set of models (Baxter and Oliver 1994). In the case of universal models, such as the SOM, the difference between model class and model set becomes unclear, because the model class is the same even though the number of parameters varies. In practice the MML community seem to favor measures based on actual implementations, while most of the MDL community seem to work in a more theoretical setting. Therefore, we have named our technique MML based. The reason is, however, based only on the practical issues, not so much on the actual methodology.

Our method is similar to Rissanen's coding approaches for non-parametric model classes, but does not require that the data distribution should be roughly uniform or the use of kernel functions. The only assumptions that need to be made are that the data values should be real numbers, and there should not be strong correlation between the variables. The first assumption is needed, because if the data was binary, for example, more efficient coding methods could be used. The second assumption is also related to coding efficiency, because if there are strong correlations, the coding method should take them into account to improve the coding. Since the data are real numbers, a quantization accuracy is needed the same way as in Rissanen's histogram approach. This accuracy, δ, can be optimized by minimizing quantization error and by maximizing coding efficiency at the same time (see Article [G]). Then, all real number data are quantized using this accuracy.

To apply the method in practice, several cluster models, m_i, created for the same data, Ω, using the same model class, $\mathcal{M}_c$ (TS-SOM), are needed. Then each

model (TS-SOM layer), and data given the model are coded into strings $C(m_i)$ and $C(\Omega \mid m_i)$, where the former contains cluster sizes, $\#(\Omega_k)$, and centroids, $\boldsymbol{w}(k)$, and the latter the residuals, $\boldsymbol{\epsilon}(j) = \boldsymbol{x}(j) - \boldsymbol{w}(c)$, where $c = \arg\min_k \|\boldsymbol{x}(j) - \boldsymbol{w}(k)\|^2$. For the coding, universal codes, such as the *log** code (Rissanen 1989), could be used. But if there is plenty of data, better results can be obtained by using optimal codes of length $-\log_2 p(\tilde{x})$, where $p(\tilde{x})$ is the frequency of quantized value $\tilde{x}$, a codebook approach (Oliver and Hand 1994), and an efficient prefix coding. Thus the coded string of each model, m_i, consists of: cluster sizes (coded integers), multidimensional cluster centroid vectors (quantized and coded using codebook), residuals (quantized and coded using codebook), and the coded description of the codebook. The model, which minimizes the total length of such message is considered to be the most suitable for the data, Ω.

Because we cannot have completely optimal coding for all parts of the message, we consider our message as an upper bound for the minimal code length. A lower bound can be obtained by removing, from the message, the non-optimal parts: the codebook and the cluster sizes. The true code length is somewhere between these two.

3.4 Discussion

Sometimes the SOM model can be selected rather easily, if a suitable visualization can be used. However, when the intra-cluster variances or inter-cluster distances of training variables have no clear meaning, the comparison of SOM models may require considerable amounts of manual work (see the textual data example in Article [H]). This is often the case when textual data or images are used as source, and some extracted features are used for SOM training.

4 ABOUT PARALLEL IMPLEMENTATION OF THE SOM

Even though the TS-SOM training algorithm, which is used in this work, is computationally light, the learning of huge data sets may take hours or days. This kind of delays cannot be tolerated in interactive applications, or in the experimentation of coding methods, for example. Algorithmic development allows the reduction of computational complexity significantly, but even faster training can be achieved with a combination of fast algorithms and parallel computers. In order to do experiments for the textual analysis, we needed a more efficient training algorithm. With the developed method, practical tests of our methodology could be done about two years before the computational power of normal PCs would have allowed it. Now the methodology can be used on a normal PC.

4.1 Parallel SOM Training Algorithms

There are *Single-Instruction-Multiple-Data* (SIMD) (Stallings 1998) type of hardware SOM implementations, such as (Mauduit et al. 1992, Melton et al. 1992, He and Çilingiroğlu 1993, Macq et al. 1993, Rüping et al. 1997), and the SIMD type CNAPS Neurocomputer, developed by Adaptive Solutions, can be programmed to train SOMs (hardware description and specific implementations can be found from (Steffens and Kunze 1995, Seiffert and Michaelis 2001)). With these systems, moderately sized networks can be trained quickly. The main problem is that the dimensionality of data, d_x, and the number of neurons in the model, N_v, are limited, because the number of neurons and input dimensionality cannot exceed what is implemented on the chip without significant performance losses. Therefore, one typically has to resort to estimating larger maps based on well trained smaller ones (Kohonen 1997, Kohonen et al. 2000). A more flexible solution of using modular maps, which allow several computational modules to be combined was presented in (Lightowler et al. 1997). In that approach the dimensionality can be increased by combining additional hardware modules.

Large maps can be constructed directly using software implementations on parallel *Multiple-Instruction-Multiple-Data* (MIMD) computers. There are two main types (Stallings 1998): tightly coupled shared memory computers and loosely coupled distributed memory systems, such as *cluster computers*. In the shared memory computers large amounts of data can be efficiently communicated between processors, which is not always true in clusters. Also the synchronization of processors needed in at least some parts of the SOM training algorithms is quicker. However, cluster computers usually contain more processors and can be built from less expensive hardware. Therefore, at least one *Single-Program-Multiple-Data* (SPMD) cluster implementation has been suggested (Schikuta and Weidmann 1997) for SOM training. Since algorithmic develop-

ment can change the complexity of some parts of the algorithm by orders of magnitude, the use of a computationally efficient algorithm is essential also for parallel implementations.

4.2 Parallel TS-SOM Training

The TS-SOM algorithm is already orders of magnitude faster than regular SOM algorithms. For large data sets, the training time of the TS-SOM can be further reduced by using Symmetric Multiprocessing (SMP) computers and our parallel algorithm, as presented in Article [F]. In this algorithm the training data and the currently trained TS-SOM layer are divided into as many parts as there are processors, p. Then, each iteration is divided into three parts, where 1. Each processor locates the closest neuron for (approximately) $1/p^{\text{th}}$ of the training data samples, 2. Each processor calculates new positions for $1/p^{\text{th}}$ of the neurons using the closest data points, and 3. Each processor updates the new positions for $1/p^{\text{th}}$ of the neurons using the neighborhood. Between the parts, the execution has to be synchronized to make sure that all processors have finished their work, because in all parts each processor also needs such results, which have been calculated by other processors in the previous part. Speed-ups of more than 7 have been achieved with 8 processors compared to a regular TS-SOM while using large data sets ($d_x \geq 100$ and $N_x \geq 50\,000$).

5 AUTHOR'S CONTRIBUTION IN THE PAPERS

All the educational problems and data sets have been provided by the Institute for Educational Research of University of Jyväskylä. Especially Prof. Pirjo Linnakylä, Prof. Päivi Häkkinen, and MSc Antero Malin have given us a lot of information and feedback. Most of the methods used have been developed as teamwork with DrTech Pasi Koikkalainen, and some ideas were discussed with PhD Erkki Häkkinen. However, all the data preprocessing methods, information theoretic coding techniques, text visualization methods, the parallel implementation, and the user interface for survey data analysis were developed by the author of the thesis. This is summarized as follows:

Article [A] contains many important ideas for the analysis of categorical data, but most of the solutions were not developed, yet.

Article [B] includes the key ideas for dividing the data set into parts, which are analyzed separately. The paper introduces the coding of categorical data using fuzzy set memberships, the calculation of group memberships, and the combining of the results from the subanalyses.

Article [C] is very similar to [B], but contains additional motivation for the use of subanalyses, and presents the idea of including textual data to the analysis model.

Article [D] develops our ideas for text clustering, which is applied in two problems: similar document detection for error diagnostics, and similar answer detection from surveys.

Article [E] combines all our methods for the educational research together. It is targeted to educational researchers, and therefore it was kept on a superficial level.

Article [F] motivates and presents our parallel implementation of the TS-SOM training and an idea how the use of massively parallel computers can be simulated.

Article [G] describes our model selection method for the TS-SOM, and presents the results obtained from applying it to the word category map (and to a toy example of function approximation).

Article [H] is the main methodological contribution including a more clear formalism, and motivation for the methods. This paper also includes the textual data analysis into the combining of results, and develops visualizations for presenting relations between the models used in subanalyses.

And finally, Article [I] describes our software implementation for survey data analysis.

6 CONCLUSION

In this thesis methodology and software tools are proposed for exploratory analysis of qualitative survey data sets. Different analysis approaches are used for categorical and textual data, but the results from the subanalyses can be combined in the end of the analysis, or selections from one submodel can be projected to all other models in order to reveal correlations. The reduction of manual work is a notable advantage compared to traditional applying of qualitative research methods, and the visual results can be more easily interpreted by educational researchers.

REFERENCES

Anderberg, M. *Cluster Analysis for Applications*. Academic Press, New York, 1973.

Baeza-Yates, R. and Ribeiro-Neto, B. *Modern Information Retrieval*. Addison-Wesley/ACM Press, Essex, 1999.

Baxter, R. and Oliver, J. *MDL and MML: Similarities and Differences*. Technical Report TR94/207, Department of Computer Science, Monash University, 1994.

Bernard, G. 'Experiments on Distributional Categorization of Lexical Items with Self-Organizing Maps'. In *Proc. WSOM'97: Workshop on Self-Organizing Maps*. Pages 304–309. Libella, Espoo, Finland, 1997.

Bishop, C. *Neural Networks for Pattern Recognition*. Oxford University Press, New York, 1995.

Bishop, C., Svensen, M., and Williams C. 'GTM: The Generative Topographic Mapping'. *Neural Computation*, Vol. 10, No. 1. Pages 215–234. 1998.

Burnham, K. and Anderson, D. *Model Selection and Inference – A Practical Information Theoretic Approach*. Springer-Verlag, New York, 1998.

Chaitin, G. 'On the Length of Programs for Computing Finite Binary Sequences'. *Journal of the Association for Computing Machinery*, Vol. 13, No. 4. Pages 547–569. 1966.

Cherkassky, V. and Mulier, F. *Learning from Data*. John Wiley and Sons, New York, 1998.

Cottrell, M., de Bodt, E., and Verleysen, M. 'A Statistical Tool to Assess the Reliability of Self-Organizing Maps'. In *Advances in Self-Organizing Maps*. Pages 7–14. Springer-Verlag, London, 2001.

Cover, T. and Thomas, J. *Elements of Information Theory*. John Wiley and Sons, New York, 1991.

Deerwester, S., Dumais, S., Furnas, G., Landauer, T., and Harshman, R. 'Indexing by Latent Semantic Analysis'. *Journal of the American Society for Information Science*, Vol. 41, No. 6. Pages 391–407. 1990.

Der, R. and Herrmann, M. 'Second-Order Learning in Self-Organizing Maps'. In *Kohonen Maps*. Pages 293–302. Elsevier Science, Amsterdam, 1999.

Dillenbourg, P. 'What Do You Mean by "Collaborative Learning"?'. In *Collaborative Learning: Cognitive and Computational Approaches*. Pages 1–19. Elsevier Science, Amsterdam, 1999.

Duda, R., Hart, P., and Stork, D. *Pattern Classification, 2nd Edition*. John Wiley and Sons, New York, 2001.

Edwards, A. *Likelihood*. Cambridge University Press, Cambridge, 1972.

Efron, B. 'Bootstrap Methods: Another Look at the Jackknife'. *Annals of Statistics*, Vol. 7, No. 1. Pages 1–26. 1979.

Fayyad, U. 'Data Mining and Knowledge Discovery: Making Sense out of Data'. *IEEE Expert*, Vol. 11, No. 5. Pages 20–25. 1996.

Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. 'The KDD Process for Extracting Useful Knowledge from Volumes of Data'. *Communications of the ACM*, Vol. 39, No. 11. Pages 27–34. 1996.

Frawley, W., Piatetsky-Shapiro, G., and Matheus., C. 'Knowledge Discovery in Databases – An Overview'. *AI Magazine*, Fall issue. Pages 57–70. 1992.

Glaser, B. and Strauss, A. *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Weinfeld and Nicolson, London, 1968.

Hansen, M. and Yu, B. 'Model Selection and the Principle of Minimum Description Length'. *Journal of the American Statistical Association*, Vol. 96, No. 454. Pages 746–774. 2001.

Hastie, T. and Stuetzle, W. 'Principal Curves'. *Journal of the American Statistical Association*, Vol. 84, No. 406. Pages 502–516. 1989.

Hastie, T., Tibshirani, R., and Friedman, J. *The Elements of Statistical Learning*. Springer-Verlag, New York, 2001.

He, Y. and Çilingiroğlu, U. 'A Charge-Based On-Chip Adaptation Kohonen Neural Network', *IEEE Transactions on Neural Networks*, Vol. 4, No. 3. Pages 462–469. 1993.

Herrmann, M., Bauer, H.-U., and Villmann, T. 'A Comparison of Topography Measures for Neural Maps'. In *Proc. WSOM'97: Workshop on Self-Organizing Maps*. Pages 274–279. Libella, Espoo, Finland, 1997.

Heskes, T. 'Energy Functions for Self-Organizing Maps'. In *Kohonen Maps*. Pages 303–315. Elsevier Science, Amsterdam, 1999.

Honkela, T. 'Comparisons of Self-Organized Word Category Maps'. In *Proc. WSOM'97: Workshop on Self-Organizing Maps*. Pages 298–303. Libella, Espoo, Finland, 1997.

Honkela, T., Pulkki, V., and Kohonen, T. 'Contextual Relations of Words in Grimm Tales Analyzed by Self-Organizing Maps'. In *Proc. ICANN'95:* 5^{th} *International Conference on Artificial Neural Networks*. Pages 3–7. EC2 & CIE, Paris, 1995.

Honkela, T., Kaski, S., Lagus, K., and Kohonen, T. 'WEBSOM – Self-Organizing Maps of Document Collections'. In *Proc. WSOM'97: Workshop on Self-Organizing Maps*. Pages 310–315. Libella, Espoo, Finland, 1997.

Horvitz, D. and Thompson, D. 'A Generalization of Sampling Without Replacement From a Finite Universe'. *Journal of the American Statistical Association*, Vol. 47, No. 260. Pages 663–685. 1952.

Van Hulle, M. 'Towards an Information-theoretic Approach to Kernel-based Topographic Map Formation'. In *Advances in Self-Organizing Maps*. Pages 1–6. Springer-Verlag, London, 2001.

Hutchinson, S. 'Education and Grounded Theory'. In Sherman, Webb (eds.) *Qualitative Research in Education: Focus and Methods*. Pages 123–140. Falmer Press, London, 1988.

Hyötyniemi, H. 'Minimum Description Length (MDL) Principle and Self-Organizing Maps'. In *Proc. WSOM'97: Workshop on Self-Organizing Maps*. Pages 124–129. Libella, Espoo, Finland, 1997.

Häkkinen, E. *Design, Implementation and Evaluation of Neural Data Analysis Environment*. Diss. Jyväskylä Studies in Computing, 12. University of Jyväskylä, 2001.

Häkkinen, E. and Koikkalainen, P. 'The Neural Data Analysis Environment'. In *Proc. WSOM'97: Workshop on Self-Organizing Maps*. Pages 69–74. Libella, Espoo, Finland, 1997a.

Häkkinen, E. and Koikkalainen, P. 'SOM Based Visualization in Data Analysis'. In *Proc. ICANN'97: 7$^{\text{th}}$ International Conference on Artificial Neural Networks*. Pages 601–606. Springer-Verlag, Berlin Heidelberg, 1997b.

Jurafsky, D. and Martin, J. *Speech and Language Processing*. Prentice Hall/Pearson, Upper Saddle River, NJ, 2000.

Kaski, S., Honkela, T., Lagus, K., and Kohonen, T. 'Creating an Order in Digital Libraries with Self-Organizing Maps'. In *Proc. WCNN'96: World Congress on Neural Networks*. Pages 814–817. San Diego, CA, September 15–18, 1996.

Kaski, S. and Lagus, K. 'Comparing Self-Organizing Maps'. In *Proc. ICANN'96: 6$^{\text{th}}$ International Conference on Artificial Neural Networks*. Pages 809–814. Springer-Verlag, Berlin, 1996.

Kaski, S., Honkela, T., Lagus, K., and Kohonen, T. 'WEBSOM – Self-Organizing Maps of Document Collections'. *Neurocomputing*, Vol. 21. Pages 101–117. 1998.

Kohonen, T. 'Self-Organized Formation of Topologically Correct Feature Maps'. *Biological Cybernetics*, 43. Pages 59–69. 1982.

Kohonen, T. *Self-Organizing Maps - Second Edition*. Springer-Verlag, Heidelberg, 1997.

Kohonen, T. 'Self-Organization of Very Large Document Collections: State of the Art'. In *Proc. ICANN'98:* 8^{th} *International Conference on Artificial Neural Networks, Vol. 1*. Pages 65–74. Springer-Verlag, London, 1998.

Kohonen, T., Kaski, S., Lagus, K., and Honkela, T. 'Very Large Two-level SOM for the Browsing of Newsgroups'. In *Proc. ICANN'96:* 6^{th} *International Conference on Artificial Neural Networks*. Pages 269–274. Springer-Verlag, Berlin, 1996.

Kohonen, T., Kaski, S., Lagus, K., Salojärvi, J., Honkela, J., Paatero, V., and Saarela, A. 'Self Organization of a Massive Text Document Collection'. In *Kohonen Maps*. Pages 171–182. Elsevier Science, Amsterdam, 1999.

Kohonen, T., Kaski, S., Lagus, K., Salojärvi, J., Honkela, J., Paatero, V., and Saarela, A. 'Organization of a Massive Document Collection'. *IEEE Transactions on Neural Networks*, Vol. 11, No. 3. Pages 574–585. 2000.

Koikkalainen, P. 'Fast Deterministic Self-Organizing Maps', In *Proc. ICANN'95:* 5^{th} *International Conference on Artificial Neural Networks*. Pages 63–68. EC2 & CIE, Paris, 1995.

Koikkalainen, P. 'Tree-Structured Self-Organizing Maps'. In *Kohonen Maps*. Pages 121–130. Elsevier Science, Amsterdam, 1999.

Koikkalainen, P. and Oja, E. 'Self-Organizing Hierarchical Feature Maps'. In *Proc. IJCNN'90: International Joint Conference on Neural Networks*. Pages 279–284. IEEE Press, 1990.

Kolmogorov, A. 'Logical Basis for Information Theory and Probability Theory'. *IEEE Transactions on Information Theory*, Vol. 14. Pages 662–664. 1968.

Kullback, S. and Leibler, A. 'On Information and Sufficiency'. *Annals of Mathematical Statistics*, Vol. 22, No. 1. Pages 79–86. 1951.

Kurimo, M. and Lagus, K. 'An Efficiently Focusing Large Vocabulary Language Model'. In *Proc. ICANN 2002: International Conference on Artificial Neural Networks*. Pages 1068–1073. Springer-Verlag, Berlin Heidelberg, 2002.

Lagus, K. 'Map of WSOM'97 Abstracts – Alternative Index'. In *Proc. WSOM'97: Workshop on Self-Organizing Maps*. Pages 368–372. Libella, Espoo, Finland, 1997.

Lagus, K. 'Generalizability of the WEBSOM Method to Document Collections of Various Types'. In *Proc. EUFIT'98:* 6^{th} *European Congress on Intelligent Techniques & Soft Computing, Vol. 1*. Pages 210–214. Verlag Mainz, Aachen, Germany, 1998.

Lagus, K., Honkela, T., Kaski, S., and Kohonen, T. 'Self-Organizing Maps of Document Collections: A New Approach to Interactive Exploration'. In *Proc. 2nd International Conference on Knowledge Discovery & Data Mining*. Pages 238–243. AAAI Press, Menlo Park, CA, 1996.

Lagus, K. and Kaski, S. 'Keyword Selection Method for Characterizing Text Document Maps'. In *Proc. ICANN'99: 9th International Conference on Artificial Neural Networks, Vol. 1*. Pages 371–376. IEE Press, London, 1999.

Lagus, K., Honkela, T., Kaski, S., and Kohonen, T. 'WEBSOM for Textual Data Mining'. *Artificial Intelligence Review*, Vol. 13, No. 5/6. Pages 345–364. 1999.

Lightowler, N., Spracklen, C., and Allen, A. 'A Modular Approach to Implementation of the Self-Organising Map'. In *Proc. WSOM'97: Workshop on Self-Organizing Maps*. Pages 130–135. Libella, Espoo, Finland, 1997.

Linnakylä, P. 'Quality of School Life in the Finnish Comprehensive School: A Comparative View'. *Scandinavian Journal of Educational Research*, Vol. 40, No. 1. Pages 69–85. 1996.

Linnakylä, P. and Brunell, V. 'Quality of school life in the Finnish- and Swedish-speaking schools in Finland'. In *Reading Literacy in an International Perspective*. Pages 203–222. US Department of Education, Washington, DC, 1997.

Linnakylä, P. and Malin, A. 'Profiling Students on School Satisfaction by Neural Networks'. Presented at ECER 97, Frankfurt am Main, September 24–27, 1997.

Linnakylä, P. and Malin, A. 'Exploring National and Individual Profiles in the Quality of School Life by Neural Networks'. Presented at ECER 98, Ljubljana, Slovenia, September 17–20, 1998.

Macq, D., Verleysen, M., Jespers, P., and Legat. J. 'Analog Implementation of a Kohonen Map with On-Chip Learning'. *IEEE Transactions on Neural Networks*, Vol. 4, No. 3. Pages 456–461. 1993.

Malin, A. and Linnakylä, P. 'Multilevel Modelling in Repeated Measures of the Quality of Finnish School Life'. *Scandinavian Journal of Educational Research*, Vol. 45, No. 2. Pages 145–166. 2001.

Manning, C. and Schütze, H. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, 1999.

Mauduit, N., Duranton, M., Gobert, J., and Sirat, J. 'Lneuro 1.0: A Piece of Hardware LEGO for Building Neural Network Systems'. *IEEE Transactions on Neural Networks*, Vol. 3, No. 3. Pages 414–421. 1992.

Melton, M., Phan, T., Reeves, D., and Van den Bout, D. 'The TInMANN VLSI Chip'. *IEEE Transactions on Neural Networks*, Vol. 3, No. 3. Pages 375–384. 1992.

Merkl, D. 'Content-based Document Classification with Highly Compressed Input Data'. In *Proc. ICANN'95: 5th International Conference on Artificial Neural Networks*. Pages 239–244. EC2 & CIE, Paris, 1995.

Merkl, D. 'Lessons Learned in Text Document Classification'. In *Proc. WSOM'97: Workshop on Self-Organizing Maps*. Pages 316–321. Libella, Espoo, Finland, 1997.

Merkl, D. 'Exploration of Text Collections with Hierarchical Feature Maps'. In *Proc. SIGIR'97: ACM SIGIR Conference on Research and Development in Information Retrieval*. Pages 186–195. ACM Press, New York, 1997.

Merkl, D. 'Document Classification with Self-Organizing Maps'. In *Kohonen Maps*. Pages 183–195. Elsevier Science, Amsterdam, 1999.

Merkl, D. and Rauber, A. 'Automatic Labeling of Self-Organizing Maps for Information Retrieval'. In *Proc. ICONIP'99: 6th International Conference on Neural Information Processing, Vol. 1*. Pages 37–42. IEEE Press, 1999.

Miikkulainen, R. 'Script Recognition with Hierarchical Feature Maps'. *Connection Science*, 2. Pages 83–101. 1990.

Miikkulainen, R. *Subsymbolic Natural Language Processing: An Integrated Model of Scripts, Lexicon, and Memory*. MIT Press, Cambridge, MA, 1993.

Miikkulainen, R. 'Text and Discourse Understanding: The DISCERN System'. In *A Handbook of Natural Language Processing: Techniques and Applications for the Processing of Language as Text*. Pages 905–919. Marcel Dekker, New York, 2000.

Miles, M. and Huberman, A. *Qualitative Data Analysis: An Expanded Sourcebook – 2nd Edition*. SAGE Publications, 1994.

Oliver, J. and Hand, D. *Introduction to Minimum Encoding Inference*. Technical Report TR94/205, Department of Computer Science, Monash University, 1994.

Polani, D. 'Organization Measures for Self-Organizing Maps'. In *Proc. WSOM'97: Workshop on Self-Organizing Maps*. Pages 280–285. Libella, Espoo, Finland, 1997.

Polani, D. 'On the Optimization of Self-Organizing Maps by Genetic Algorithms'. In *Kohonen Maps*. Pages 157–169. Elsevier Science, Amsterdam, 1999.

Rauber, A. and Merkl, D. 'Creating an Order in Distributed Digital Libraries by Integrating Independent Self-Organizing Maps'. In *Proc. ICANN'98: 8th International Conference on Artificial Neural Networks, Vol. 2*. Pages 773–778. Springer-Verlag, London, 1998.

Rauber, A. and Merkl, D. 'The SOMLib Digital Library System'. In *Proc. ECDL'99: European Conference on Research and Advanced Technology for Digital Libraries*. Pages 323–342. Springer-Verlag, Berlin Heidelberg, 1999.

Rissanen, J. 'Stochastic Complexity'. *Journal of the Royal Statistical Society, Series B*, Vol. 49, No. 3. Pages 223-239. 1987.

Rissanen, J. *Stochastic Complexity in Statistical Inference*. World Scientific, Singapore, 1989.

Ritter, H. and Kohonen, T. 'Self-Organizing Semantic Maps'. *Biological Cybernetics*, Vol. 61, No. 4. Pages 241–254. 1989.

Rüping, S., Porrmann, R., and Rückert, U. 'SOM Hardware-Accelerator'. In *Proc. WSOM'97: Workshop on Self-Organizing Maps*. Pages 136–141. Libella, Espoo, Finland, 1997.

Salton, G. and McGill, M. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.

Sammon Jr., J. 'A Nonlinear Mapping for Data Structure Analysis'. *IEEE Transactions on Computers*, Vol. 18. Pages 401–409. 1969.

Schikuta, E. and Weidmann, C. 'Data Parallel Simulation of Self-Organizing Maps on Hypercube Architectures'. In *Proc. WSOM'97: Workshop on Self-Organizing Maps*. Pages 142–147. Libella, Espoo, Finland, 1997.

Schwarz, G. 'Estimating the Dimension of a Model'. *Annals of Statistics*, Vol. 6. Pages 461–464. 1978.

Sebastiani, F. 'Machine Learning in Automated Text Categorization'. *ACM Computing Surveys*, Vol. 34, No. 1. Pages 1–47. 2002.

Seiffert, U. and Michaelis, B. 'Multi-dimensional Self-Organizing Maps on Massively Parallel Hardware'. In *Advances in Self-Organising Maps*. Pages 160–166. Springer-Verlag, London, 2001.

Shannon, C. 'A Mathematical Theory of Communication'. *Bell Systems Technical Journal*, 47. Pages 143–157. 1948.

Stallings, W. *Operating Systems - Internals and Design Principles, Third Edition*. Prentice Hall, Upper Saddle River, NJ, 1998.

Steffens, J. and Kunze, M. 'Implementation of the Supervised Growing Cell Structure on the CNAPS Neurocomputer'. In *Proc. ICANN'95:* 5^{th} *International Conference on Artificial Neural Networks*. Pages 51–56. EC2 & CIE, Paris, 1995.

Stone, M. 'Cross-Validatory Choice and Assessment of Statistical Predictions'. *Journal of the Royal Statistical Society, Series B*, Vol. 36, No. 2. Pages 111–147. 1974.

Tesch, R. *Qualitative Research: Analysis Types and Software Tools*. Falmer Press, Hampshire, UK, 1990.

Thurstone, L. *Multiple-factor Analysis: a Development and Expansion of the Vectors of Mind*. University of Chicago Press, Chicago, 1949.

Tukey, J. 'The Future of Data Analysis'. *Annals of Mathematical Statistics*, 33. 1962.

Tukey, J. *Exploratory Data Analysis*. Addison-Wesley, Reading, MA, 1977.

Vapnik, V. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

Vesanto, J. 'SOM-based Data Visualization Methods'. *Intelligent Data Analysis*, Vol. 3, No. 2. Pages 111–126. 1999.

Villmann, T. 'Topology Preservation in Self-Organizing Maps'. In *Kohonen Maps*. Pages 279–292. Elsevier Science, Amsterdam, 1999.

Villmann, T. and Bauer, H. 'The GSOM-Algorithm for Growing Hypercubical Output Spaces in Self-Organizing Maps'. In *Proc. WSOM'97: Workshop on Self-Organizing Maps*. Pages 286–291. Libella, Espoo, Finland, 1997.

Wallace, C. and Boulton, D. 'An Information Measure for Classification'. *Computer Journal*, Vol. 11. Pages 185–194. 1968.

Wallace, C. and Freeman, P. 'Estimation and Inference by Compact Coding'. *Journal of the Royal Statistical Society, Series B*, Vol. 49, No. 3. Pages 240–265. 1987.

Williams, T. and Batten, M. *The Quality of School Life*. ACER Research Monograph No. 12, Hawthorn, Victoria, 1981.

Williams, T. and Roey, S. 'Consistencies in the quality of school life'. In *Reading Literacy in an International Perspective*. Pages 193–202. US Department of Education, Washington, DC, 1997.

Zadeh, L. 'Fuzzy Sets'. *Information and Control*, Vol. 8. Pages 338–353. 1965.

Zimmermann, H.-J. *Fuzzy Set Theory – and Its Applications*. Kluwer, Hingham, MA, 1985.

Zrehen, S. 'Analyzing Kohonen Maps with Geometry'. In Proc. *ICANN'93:* 3rd *International Conference on Artificial Neural Networks*. Pages 609–612. Springer, London, 1993.

YHTEENVETO (FINNISH SUMMARY)

Tässä tutkimuksessa on kehitetty laskennallisesti älykkäitä menetelmiä koulutuksen tutkimuksen kyselytutkimuksiin liittyvien tietoaineistojen analyysiä varten. Menetelmillä on mahdollista käsitellä monimutkaisia ja eri tietotyyppejä sisältäviä tietoaineistoja siten, että eri tietotyypit käsitellään omissa alianalyyseissä, joiden tulokset voidaan lopuksi yhdistää. Menetelmäkehys mahdollistaa samankaltaisten havaintojen ryhmien löytämisen, ryhmien identifioinnin taustatiedon avulla, eri alianalyyseistä valittujen ryhmien vertaamisen, eri populaatioista valittujen otosten vertaamisen, ja samankaltaisten tekstivastausten löytämisen. Työn yhteydessä on kehitetty myös ohjelmisto, joka sisältää kehitetyt menetelmät, ja jolla aineistoja on tutkittu.

JYVÄSKYLÄ STUDIES IN COMPUTING

1 ROPPONEN, JANNE, Software risk management - foundations, principles and empirical findings. 273 p. Yhteenveto 1 p. 1999.
2 KUZMIN, DMITRI, Numerical simulation of reactive bubbly flows. 110 p. Yhteenveto 1 p. 1999.
3 KARSTEN, HELENA, Weaving tapestry: collaborative information technology and organisational change. 266 p. Yhteenveto 3 p. 2000.
4 KOSKINEN, JUSSI, Automated transient hypertext support for software maintenance. 98 p. (250 p.) Yhteenveto 1 p. 2000.
5 RISTANIEMI, TAPANI, Synchronization and blind signal processing in CDMA systems. - Synkronointi ja sokea signaalinkäsittely CDMA järjestelmässä. 112 p. Yhteenveto 1 p. 2000.
6 LAITINEN, MIKA, Mathematical modelling of conductive-radiative heat transfer. 20 p. (108 p.) Yhteenveto 1 p. 2000.
7 KOSKINEN, MINNA, Process metamodelling. Conceptual foundations and application. 213 p. Yhteenveto 1 p. 2000.
8 SMOLIANSKI, ANTON, Numerical modeling of two-fluid interfacial flows. 109 p. Yhteenveto 1 p. 2001.
9 NAHAR, NAZMUN, Information technology supported technology transfer process. A multi-site case study of high-tech enterprises. 377 p. Yhteenveto 3 p. 2001.
10 FOMIN, VLADISLAV V., The process of standard making. The case of cellular mobile telephony. - Standardin kehittämisen prosessi. Tapaustutkimus solukkoverkkoon perustuvasta matkapuhelintekniikasta. 107 p. (208 p.) Yhteenveto 1 p. 2001.
11 PÄIVÄRINTA, TERO, A genre-based approach to developing electronic document management in the organization. 190 p. Yhteenveto 1 p. 2001.
12 HÄKKINEN, ERKKI, Design, implementation and evaluation of neural data analysis environment. 229 p. Yhteenveto 1 p. 2001.
13 HIRVONEN, KULLERVO, Towards Better Employment Using Adaptive Control of Labour Costs of an Enterprise. 118 p. Yhteenveto 4 p. 2001.
14 MAJAVA, KIRSI, Optimization-based techniques for image restoration. 27 p. (142 p.) Yhteenveto 1 p. 2001.
15 SAARINEN, KARI, Near infra-red measurement based control system for thermo-mechanical refiners. 84 p. (186 p.) Yhteenveto 1 p. 2001.
16 FORSELL, MARKO, Improving Component Reuse in Software Development. 169 p. Yhteenveto 1 p. 2002.
17 VIRTANEN, PAULI, Neuro-fuzzy expert systems in financial and control engineering. 245 p. Yhteenveto 1 p. 2002.
18 KOVALAINEN, MIKKO, Computer mediated organizational memory for process control. Moving CSCW research from an idea to a product. 57 p. (146 p.) Yhteenveto 4 p. 2002.
19 HÄMÄLÄINEN, TIMO, Broadband network quality of service and pricing. 140 p. Yhteenveto 1 p. 2002.
20 MARTIKAINEN, JANNE, Efficient solvers for discretized elliptic vector-valued problems. 25 p. (109 p.) Yhteenveto 1 p. 2002.
21 MURSU, ANJA, Information systems development in developing countries. Risk management and sustainability analysis in Nigerian software companies. 296 p. Yhteenveto 3 p. 2002.
22 SELEZNYOV, ALEXANDR, An anomaly intrusion detection system based on intelligent user recognition. 186 p. Yhteenveto 3 p. 2002.
23 LENSU, ANSSI, Computationally intelligent methods for qualitative data analysis. 57 p. (180 p.) Yhteenveto 1 p. 2002.