Helsinki University of Technology
Dissertations in Computer and Information Science
Espoo 2004 Report D5

From insights to innovations: data mining, visualization, and user interfaces

Johan Himberg

Dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the Department of Computer Science and Engineering for public examination and debate in Auditorium AS1 at Helsinki University of Technology (Espoo, Finland) on the 5th of November, 2004, at 12 o'clock noon.

Helsinki University of Technology
Department of Computer Science and Engineering
Laboratory of Computer and Information Science
P.O.Box 5400
FIN-02015 HUT
FINLAND

Distribution:
Helsinki University of Technology
Laboratory of Computer and Information Science
P.O.Box 5400
FIN-02015 HUT
FINLAND
Tel. +358 9 4513 272
Fax +358 9 4513 277
http://www.cis.hut.fi
Available in PDF format at http://lib.hut.fi/Diss/2004/isbn9512273373

ISBN 951-22-7336-5 (printed version)
ISBN 951-22-7337-3 (electronic version)
ISSN 1479-7020

Otamedia Oy
Espoo 2004

Himberg, J. (2004): **From insights to innovations: data mining, visualization, and user interfaces**. Doctoral thesis, Helsinki University of Technology, Dissertations in Computer and Information Science, Report D5, Espoo, Finland.

Keywords: adaptive user interface, adaptive keyboard, context-awareness, data mining, independent component analysis, information visualization, mobile-awareness, proximity preserving projection, self-organizing map, sensor fusion, time series segmentation, visual data mining

ABSTRACT

This thesis is about data mining (DM) and visualization methods for gaining insight into multidimensional data. Novel, exploratory data analysis tools and adaptive user interfaces are developed by tailoring and combining existing DM and visualization methods in order to advance in different applications.

The thesis presents new visual data mining (VDM) methods that are also implemented in software toolboxes and applied to industrial and biomedical signals: First, we propose a method that has been applied to investigating industrial process data. The self-organizing map (SOM) is combined with scatterplots using the traditional color linking or interactive brushing. The original contribution is to apply color linked or brushed scatterplots and the SOM to visually survey local dependencies between a pair of attributes in different parts of the SOM. Clusters can be visualized on a SOM with different colors, and we also present how a color coding can be automatically obtained by using a proximity preserving projection of the SOM model vectors. Second, we present a new method for an (interactive) visualization of cluster structures in a SOM. By using a contraction model, the regular grid of a SOM visualization is smoothly changed toward a presentation that shows better the proximities in the data space. Third, we propose a novel VDM method for investigating the reliability of estimates resulting from a stochastic independent component analysis (ICA) algorithm. The method can be extended also to other problems of similar kind. As a benchmarking task, we rank independent components estimated on a biomedical data set recorded from the brain and gain a reasonable result.

We also utilize DM and visualization for mobile-awareness and personalization. We explore how to infer information about the usage context from features that are derived from sensory signals. The signals originate from a mobile phone with on-board sensors for ambient physical conditions. In previous studies, the signals are transformed into descriptive (fuzzy or binary) context features. In this thesis, we present how the features can be transformed into higher-level patterns, contexts, by rather simple statistical methods: we propose and test using minimum-variance cost time series segmentation, ICA, and principal component analysis (PCA) for this purpose. Both time-series segmentation and PCA revealed meaningful contexts from the features in a visual data exploration.

We also present a novel type of adaptive soft keyboard where the aim is to obtain an ergonomically better, more comfortable keyboard. The method starts from some conventional keypad layout, but it gradually shifts the keys into new positions according to the user's grasp and typing pattern.

Related to the applications, we present two algorithms that can be used in a general context: First, we describe a binary mixing model for independent binary sources. The model resembles the ordinary ICA model, but the summation is replaced by the Boolean operator OR and the multiplication by AND. We propose a new, heuristic method for estimating the binary mixing matrix and analyze its performance experimentally. The method works for signals that are sparse enough. We also discuss differences on the results when using different objective functions in the FastICA estimation algorithm. Second, we propose "global iterative replacement" (GIR), a novel, greedy variant of a merge-split segmentation method. Its performance compares favorably to that of the traditional top-down binary split segmentation algorithm.

Foreword

During the formation of this thesis 1998–2004, I have had a privilege to work in “Two households, both alike in dignity”: the Laboratory of Computer and Information Science (CIS) at Helsinki University of Technology (HUT) and Nokia Research Center (NRC). The excellent working conditions and resources for finalizing my thesis are due to CIS—and to the Neural Networks Research Centre, established by Academician Teuvo Kohonen and presently led by Academy Professor Erkki Oja. A part of the research was carried out in technology programs of the National Technology Agency of Finland (Tekes); the collaborating private companies are acknowledged in the publications. Finalizing this thesis has also been supported by a personal grant from Nokia Foundation which I gratefully acknowledge.

I sincerely thank my good old friend, Docent Aapo Hyvärinen, the tutor of this thesis, for all guidance, cheerful encouragement, and constructive criticism—and for moments of laughter. Professor Olli Simula, the supervisor of my thesis, has been very supportive and often saved my day with his sunny disposition and trust in my work. Aapo and Olli, as well as Academy Professor Heikki Mannila and Professor Jaakko Hollmén all have devoted a great deal of their time, world-class expertise, and sources of sympathy, to guide and support me during these years. I’m deeply indebted to all of these gentlemen.

The reviewers of this thesis, Professor Sami Khuri at San José State University, California and Professor Olli Silvén at Oulu University, Finland, have been thorough in examining and commenting the manuscript. This has been most helpful and educational for me and, surely, valuable to the readers as an improved quality of the presentation. I am also glad that Professor Juha Röning at Oulu University has accepted the invitation to act as the opponent in the public examination and debate.

I carried out the work related to self-organizing maps in the brave, old Intelligent Data Engineering gang together with the co-authors Mr. Jussi Ahola, Dr. Esa Alhoniemi, Prof. Olli Simula, and Dr. Juha Vesanto. I studied context-awareness and personalization in the enjoyable company of co-authors Ms. Jonna Häkkilä, Mr. Petri Kangas, Mr. Kalle Korpiaho, Mr. Panu Korpipää, Prof. Heikki Mannila, Dr. Jani Mäntyjärvi, Ms. Johanna Tikanmäki, and Prof. Hannu T.T. Toivonen. And let me not forget Aapo with whom I have had the opportunity to write two publications in this thesis. While being at NRC and after that, I have been working in close co-operation with Dr. John Adrian Flanagan, and of course, Jani. Working as a team with Adrian and Jani has been both fruitful and fun, and it has enhanced my work on the area of context-awareness. I am also thankful for Dr. Olli Karonen, my boss at NRC who supported me in managing both the development work and the academic endeavors simultaneously. Working with all of these fine people has been a pleasure.

I have been fortunate to get worthy comments on different versions of the manuscript. There are many to be acknowledged for their benevolence: Dr. Sampsa Laine’s insightful comments improved the early version of the manuscript. Dr. Ella Bingham, Prof. Samuel Kaski, Ms. Anne Patrikainen, Dr. Kai Puolamäki, Mr. Timo Similä, and Mr. Jarkko Venna have given valuable comments on their fields of expertise. Adrian and my dear wife, Mrs. Marle Himberg, have read the manuscript in close detail and told where I should make the presentation clearer and closer to standard English.

Atmosphere at work has been friendly and stimulating, thanks to all of my splendid colleagues at CIS and NRC. I just spotlight the present year: Intelligent Data Engineering Group (both the Old Gang and the Next Generation) and Pattern Discovery Group have been such a jolly good company; Anne and Nikolaj have been superb company when sharing the office room; also, Antti, Ella, Heli, Jaakko H., Jouni, Jukka, Miki, Salla, and Sampsa have frequently been there for company me for lunch—or for therapeutic coffee breaks. With Jaakko S. and Ville, I have had many “peer-to-peer discussions” on preparing the manuscript and arranging the *karonkka* dinner. Ms. Leila Koivisto, Ms. Tarja Pihamaa, Mr. Markku Ranta, and Dr. Miki Sirola have gladly helped with infrastructure and bureaucracy.

During the course of this thesis, I have been involved in data mining projects that, although not reported here, have been influential in my insight of the field. Therefore, I wish to thank Mr. Pekka Hippeläinen, Dr. Pertti Huuskonen, Mr. Jukka Laine, Dr. Vesa Luiro, Dr. Mika Klemettinen, Mr. Ola Rinta-Koski, Dr. Kimmo Valkealahti, Dr. Urpo Tuomela, and Mr. Juha Päivärinta for co-operation and communications.

Finally, I wish to thank my family, relatives and friends for their support in the long run and cheering-up during the final push. I am indebted to my parents Mikael and Lea Himberg for their love and life-time support; sadly, these thanks do not reach my mother anymore. Finally, my thanks to my dear wife Marle who has been so loving, caring, and patient.

In Helsinki, 7th October 2004,

Johan Himberg

Contents

Legend

Abbreviations

AL	group average linkage
BSS	blind signal separation
CCA	curvilinear component analysis
CL	complete linkage (furthest neighbor linkage)
GIR	global iterative replacement
ICA	independent component analysis
KDD	knowledge discovery (in databases)
KNN	K nearest neighbors
LIR	local iterative replacement
MDS	multidimensional scaling
MEG	magnetoencephalograph(y)
PCA	principal component analysis
RGB	red-green-blue color space
SL	single linkage (nearest neighbor linkage)
SOM	self-organizing map
SSE	sum of squared errors
UI	user interface
VDM	visual data mining
wpm	words per minute
WWW	world-wide web
1D,2D,3D . . .	1-, 2-, 3-,. . . dimensional; 1, 2, 3, . . . dimensions

Symbols

In general, matrices are denoted by capital bold ($\mathbf{A}$) and vectors by lower-case bold ($\mathbf{a}$) Latin or Greek alphabets. A scalar element of the matrix or vector is denoted by the same letter but using lower-case, plain font type (a_{ij}).

$\mathbf{A}$	mixing matrix
$\mathbf{A}^B$	mixing matrix that has binary elements, i.e., $a_{ij}^B \in \{0,1\}$
C	elements of a data set
$\lvert C\rvert$	cardinality of set C: in this case, number of elements in a data set $= N$
C_i	elements of a cluster or segment i
$\mathbf{c}$	centroid vector (average) of a data set
$\mathbf{c}_i$	centroid vector (average) of cluster C_i; codebook vector i
$\mathbf{D}$	distance, dissimilarity, or dispersion (scatter) matrix
$\mathbf{D}_B$	between-cluster dispersion matrix
d_{ij}	dissimilarity (distance) between data items i and j; element of a dissimilarity (distance) matrix
d_{ij}^*	distance between items i and j in a projection
$\mathbf{D}_T$	total dispersion matrix
$\mathbf{D}_W$	within-cluster dispersion matrix
$d(\mathbf{x},\mathbf{y})$	dissimilarity between $\mathbf{x}$ and $\mathbf{y}$; e.g., the Euclidean distance $\lVert\mathbf{x}-\mathbf{y}\rVert=\sqrt{\sum_i(x_i-y_i)}$
$\mathbf{D}_2$	matrix whose elements are squared elements of an Euclidean distance matrix, i.e., where $d_{2ij} = d_{ij}$
Δ	within-cluster dispersion (cluster diameter) index
δ	between cluster dissimilarity, e.g., single-linkage δ_{SL}
E	expectation
$\mathbf{E}$	matrix consisting of eigenvectors
i,j,k,l	general purpose indices
I_{DB}	Davies-Bouldin clustering validity index
I_R	R-index (a clustering validity index)
$I_T(k)$	projection trustworthiness index for neighborhood size k
J	objective function of FastICA
$\mathbf{\Lambda}$	$M \times M$ diagonal matrix with diagonal elements $\lambda_1,\ldots,\lambda_M$, e.g., eigenvalues
K	number of segments or clusters
κ	general purpose normalization factor
M	number of variables, i.e., data dimension
N	number of items in a data set $= \lvert C\rvert$
O	order: $f(N) = O(h(N))$, if $\lvert f(N)/h(n)\rvert$ remains bounded when $N \to \infty$
r	linear correlation coefficient
$\mathbf{S}$	independent components in an $M \times N$ matrix
s_{ij}	similarity between objects i and j; element of a similarity matrix
σ	1) Gaussian kernel width or neighborhood width; 2) standard deviation
t	(time) index, iteration step
V_i	nearest neighbor (Voronoi) set of codebook vector $\mathbf{c}_i$
$\mathbf{W}$	demixing matrix

$\mathbf{X}$	data: an $M \times N$ matrix of column vectors $\mathbf{x}(i)$
$\mathbf{X}_0$	centered data
$\mathbf{X}^*$	projected data
$\mathbf{x}$	sample data vector, random vector
$\mathbf{x}(i)$	ith data vector (column vector) in a data set $\mathbf{X}$
$\mathbf{x}^*(i)$	projected data vector
$\mathbf{Z}$	whitened data

Chapter 1

Introduction

1.1 Course of this thesis

1.1.1 Seeds—diverse problems

The seeds of this thesis are data-analysis problems in applied research that I have encountered both in academia and industry. A part of the work is conducted in Helsinki University of Technology, in a research group with interest in analyzing and visualizing complex data, especially from metal and pulp process industry. The aim of the work is to provide exploratory data analysis and visualization tools for analyzing and monitoring industrial processes. The economical and environmental interests related to process analysis are quite obvious. Furthermore, I extend the use of visualization in assessing the results of an analysis performed on biomedical signals recorded from the human brain.

Another part of this thesis is related to my duties at Nokia Research Center. The research is inspired by a vision to make a mobile device context-aware and more personal. Context-awareness means that devices and applications should adapt favorably to the usage situation. The concept of context-awareness is relatively young in mobile communications and human-computer interaction, but interest in it grows rapidly due to the increasing mobility and level of sophistication of communication devices, computers, and services. To set a futuristic goal, the mobile devices could use commonsense reasoning, learn the facts of life and "learn to live" with their users in changing contexts.

As a result, this thesis touches on three interesting but quite diverse applications with different aims:

- detecting the cause of problems in an industrial process,
- analyzing biomedical signals, and
- personalizing a mobile phone user interface by making it context-aware.

Now, what is the scope of my thesis? And what is the scientific relevance of it? In the following subsection, I will shortly answer to the first question by explaining the main themes of my thesis on a general level. After that, in Sec. 1.1.3, I summarize the novel, scientific contributions of this thesis.

1.1.2 Growth—the storyline

Scope

Process industry and brain research are intriguing subjects but they are not an end in themselves in this thesis. The subject matter of this thesis is to investigate and utilize the signal data in general, in an exploratory manner. The methods of visual and exploratory data analysis, i.e., the visual data mining, are also related to human-computer interaction studies. Typically, some kind of a user interface is used in controlling a data mining process and in showing its results. My work is also about an other kind of combination of data mining and user interfaces: methods of data mining are used to build new, adaptive user interface applications. In the following, I rough out the common factors, the three-folded yarn of this thesis, in more detail.

Data mining on multidimensional sensor data

Firstly, in each application there are multisensor measurements from a complex system. Understanding multidimensional data is difficult and often requires sophisticated, computerized tools of analysis and presentation. Extracting useful patterns from such data and making them understandable for the analyst, are problems encountered in disciplines called knowledge discovery in databases and data mining.

Thus, the applications are knitted together by the goal of obtaining insight into the sensor data. In process industry, the measurements consist of process parameters that may be controllable or just monitored. In brain research, signals are supplied by a magnetoencephalograph (MEG), a sophisticated instrument that is capable of measuring the magnetic fields produced by the brain in sixty locations on the scalp. In the study relating to context-awareness, the data consists of various simultaneous features describing the user's environment. These features are derived from sensors monitoring the ambient conditions of a mobile device and the acceleration of the device itself.

Visualization and human-computer interaction

Secondly, in a major part of my thesis, a user interface or visualization is combined with computational methods of data mining. Exploring the patterns in data is advanced by using visualization techniques. Visualization means computer supported visual representations of data. Extensive use of visualization in data mining is called visual data mining, which sets the framework for my thesis. The aim of information visualization is to create interactive "windows" to the data. Visual data mining is not only about presenting the results of the data analysis, but about exposing the data and the data mining method transparently to the analyst. This promotes insight both into the data and the analysis itself.

This thesis also deals with applications where a user interface is changed according to the detected patterns in data. This applies to the mobile user interface applications where the aim is to make the mobile device sensitive to the user actions and context, i.e., to make it context-aware. Thus, this thesis touches human-computer interaction not only through the interactive nature of visual data mining but also through some of the applications.

Research and development on novel application areas

Thirdly, the novelty of the application domains characterizes this work. To begin with, in order to gain understanding in the first place, one has to get some overall insight into the data produced by the sensor system. This might be called "playing with the data" or exploring the data. This is where the visual data mining or visual data exploration are of use. Data exploration is a necessary phase in order to have some ideas for further research.

Commercial competition encourages the realization of even vague ideas in novel application areas. This can be clearly seen in the context-aware mobile phone applications which compete on the cutting edge of technology. The aim there is to sketch a prototypic system to demonstrate that some application indeed could be constructed. Tangible results from materializing the ideas are the applied patents.

Novelty of a method or a research area may hinder obtaining results in academic research as well. In this thesis, the analysis of biomedical data exemplifies this problem. There, the problem lies in assessing the results provided by a stochastic algorithm on a limited sample. How to do this in a well-established manner, is an important research question. However, from a practitioner's standpoint, any tool capable of providing insight into the problem helps in advancing the real matter of interest: finding interesting signals from the brain or other systems. An exploratory data analysis approach combined with visualization provides a fairly good way of obtaining results which are generalizable into other similar problems as well.

Main theme

To sum up the theme of this thesis, I concentrate on the rapid deployment of methods for gaining insight into multidimensional data. This is done in order to promote research and development in new application areas. This involves developing novel, exploratory data analysis tools and adaptive user interfaces by tailoring and combining existing data analysis methods and visualizations. The visualizations also provide insight into the results and characteristics of the data mining methods themselves.

1.1.3 Fruits—scientific and practical relevance

In this section, I present the main contributions of this thesis, the background, and motivation for them. Section 1.1.4 re-describes the contributions in detail by publication and author at a time.

Scientific contributions

The first part of my thesis is about visual data mining methods. To begin with, it regards the self-organizing map (SOM) as a widely accepted and established method on (visual) data mining and proposes new options for its visualization. I present new methods that utilize color linking, brushing, and interactive projection in enhancing SOM visualizations and in linking it into other representations of the data.

An established way of visualizing SOM, the component planes, gives a good general view to attribute values in different parts of the SOM. However, it is not easy to see details of the pairwise dependencies between the attributes in it. Meanwhile, an ordinary scatterplot is an easily conceivable means of investigating such dependencies. The first original contribution is a method where the general view mediated by the component planes is combined with selected scatterplots and, possibly, a time series plot of data using color linking or interactive brushing. The method is applied for industrial process data analysis in order to visually survey, simultaneously, dependencies in and clustering of the data, as well as the relation to the original, multidimensional time series.

Another problem with the SOM visualization has to do with detecting clusters in data. Although, the SOM aims at retaining the topology of the data, the regular grid of the SOM units in the output space does not, in itself, visualize clusters in the data space. Different solutions exist: Contour lines or shades of gray on the SOM grid can be used to visualize the mutual distances between model vectors (in the data space) of the neighboring SOM units. Clusters can also be visualized on a SOM with different colors. I propose how the color coding can automatically be obtained by using a proximity preserving projection of the SOM model vectors. (The color coding can then be used to link the SOM to other representations, like scatterplots.)

Another option for highlighting the clusters is to reflect the distances between the model vectors of the neighboring SOM units in the input space by the distances in the output space visualization. Different solutions exist: One can rigorously optimize a reasonable cost function that sets the inter-unit distances in the visualization to correspond those in the input data space. Alternatively, the are variants of the SOM where unit locations in the output space inherently try to reflect the shape (and cluster structure) of the input data.

I take a different approach and present a novel (interactive) method for visualization of cluster structures in a SOM. The method does not aim at a rigor optimization of a cost function but easily computes a sequence of projections by repeatedly applying a simple contraction model. As a consequence, the visualization can be smoothly changed, back and forth, between the regular grid, a presentation that reflects the inter-unit proximities in the data space, and a densely clustered view. The same method can, eventually, be applied to any initial proximity preserving projection.

To continue with visual data mining, my thesis addresses a problem of assessing results from a stochastic estimation algorithm by resampling, clustering, and visualization: FastICA is an established algorithm, with many favorable properties, for extracting independent components. A feature of FastICA is that its optimization procedure may end to different results for the same data depending on the initial conditions of the estimation. This is a problem similar to many practical and widely used, greedy algorithms, as e.g., K-means.

How to assess the reliability of the results on a sample, is a general problem in independent component analysis (ICA). A resampling method has been presented to solve this, but applying it directly to FastICA without changes may cause difficulties. As a solution, I propose, with my co-author, a novel method, *Icasso*, for investigating the reliability of estimates resulting from FastICA. *Icasso* is based on bottom-up clustering of resampled ICA estimates. Furthermore, we propose a proximity preserving visualization of the estimate space for additional check up and further investigation of the clustering results. The method can be extended also to other problems of similar kind, say, assessing the results of the K-means algorithm.

The second part of my thesis is joint work about utilizing machine learning in context recognition, mobile-awareness and personalization. We explore how to infer information about the usage context from features that are derived from sensory signals. The signals originate from a mobile phone equipped with on-board sensors for ambient physical conditions: touch, noise, light, temperature, humidity and acceleration of the device itself. In previous studies, the signals are transformed into descriptive (fuzzy or binary) context features.

Earlier, Hidden Markov models, self-organizing maps, and naïve Bayesian classifiers have been proposed to be used, especially, for the purpose of supervised context recognition. We present how the features can be transformed into higher-level patterns, contexts, by rather simple statistical methods that seem not to be previously appeared in connection to context recognition: we propose and test using minimum-variance cost time series segmentation for detecting context-changes in time—and principal component analysis (PCA) and ICA for extracting contexts as latent sources. Time series segmentation is an attractive approach since there are computationally light, though suboptimal, algorithms available. PCA and ICA could extract contexts unsupervisedly as latent variables. In contrast to the previous approaches, where only one context is active at a time, several latent variables can be active simultaneously. Time series segmentation and PCA revealed meaningful contexts from the features in a visual data exploration. In this case, ICA did not provide that useful statistical aggregates for the test data, and I will discuss the reasons for this in this thesis.

Related to the context-aware applications, we also present two algorithms that can be used in a general context: the first one is related to latent variable models and the second one to time series segmentation: First, our test data in context-awareness studies consists of multidimensional, (almost) binary data. We assumed that the data could, in general, emerge from a non-linear, Boolean mixing model of independent binary sources, and possibly binary noise. The prior work with ICA and binary data has concentrated on linear mixtures of discrete (binary) sources and Gaussian noise, which was not applicable model in the problem. To advance rapidly and to take use of the effective methods for linear ICA, we propose a new, heuristic method for estimating the binary mixing matrix of the model and analyzed its performance experimentally. The method works for signals that are sparse enough. We also discuss differences on the results when using different objective functions in the FastICA estimation algorithm. Second, we propose "global iterative replacement" (GIR), a novel, greedy and fast variant of a merge-split time series segmentation method. Its performance compares favorably to that of the traditional top-down binary split segmentation algorithm.

Finally, we present a novel type of an adaptive soft keyboard. The prior work on optimized, soft keyboards typically aims for one common layout where the keys are completely relocated to an optimal configuration for fast typing. This is done according to general theoretical models of language and human motor system. We propose reshaping the keyboard to feel personal and present an adaptive keyboard that subtly changes the keyboard to fit the user's individual typing pattern. The individual physiology of the hand and the personal fashion of typing are not captured by the generally used models in keyboard optimization. In the absence of a well-established model, we started exploring the problem based on a very simple learning technique: the method starts from a given keypad layout and shifts the keys into new positions according to the user's grasp using a simple adaptation rule. Preliminary, quantitative results on user satisfaction and adaptation results are encouraging

Other contributions

What comes to the practical import, one aspect of my thesis is to demonstrate the rapid deployment of data mining into new applications in mobile user interfaces for insights and innovations. In fact, the research and development work done while preparing this thesis has led to some patent applications.[1] More importantly, some of the results have been concretized also in two freeware software packages the *SOM Toolbox* and *Icasso*. These packages are targeted for researchers both in academia and industry.

[1] Published applications include WO03043356: A method for controlling operation of a mobile device by detecting usage situations, WO03107168: Electronic device and method of managing its keyboard, and U.S. Pat. Appl. 20040002948: Portable electronic device and method for determining its context.

1.1.4 Publications and their original features

The original features of my thesis are contained in eight publications. Seven of them are contained in peer reviewed proceedings of international scientific conferences and one is an invited, edited book chapter. The list below gives the numbering that is used when referring to the publications. The order of the publications is by topic which is explained below in the description of the contents and original features of the publications, and my personal contributions to them.

1. Johan Himberg, Jussi Ahola, Esa Alhoniemi, Juha Vesanto, and Olli Simula (2001). The Self-Organizing Map as a Tool in Knowledge Engineering. *Pattern Recognition in Soft Computing Paradigm*. (ed. Nikhil R. Pal). World Scientific Publishing, pages 38–65.

2. Johan Himberg (1998). Enhancing the SOM Based Data Visualization by Linking Different Data Projections. In *Proceedings of the International Symposium on Intelligent Data Engineering and Learning (IDEAL'98)*, Hong Kong, pages 427–434.

3. Johan Himberg (2000). A SOM Based Cluster Visualization and Its Application for False Coloring. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN2000)*, Como, Italy, vol. 3, pages 587–592.

4. Johan Himberg and Aapo Hyvärinen (2003). Icasso: Software for investigating the reliability of ICA estimates by clustering and visualization. In *Proceedings of the IEEE International Workshop on Neural Networks for Signal Processing (NNSP'03)*, Toulouse, France, pages 259–268.

5. Johan Himberg, Jani Mäntyjärvi and Panu Korpipää (2001). Using PCA and ICA for Exploratory Data Analysis in Situation Awareness. In *Proceedings of the IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI2001)*, Baden-Baden, Germany, pages 127–131.

6. Johan Himberg and Aapo Hyvärinen (2001). Independent component analysis for binary data: An experimental study. In *Proceedings of the International Conference on Independent Component Analysis and Blind Signal Separation (ICA2001)*, San Diego, California, pages 552–556.

7. Johan Himberg, Kalle Korpiaho, Heikki Mannila, Johanna Tikanmäki, and Hannu T.T. Toivonen (2001). Time series segmentation for context recognition in mobile devices. In *Proceedings of the IEEE International Conference on Data Mining (ICDM2001)*, San José, California, pages 203–210.

8. Johan Himberg, Jonna Häkkilä, Jani Mäntyjärvi, and Petri Kangas (2003). On-line personalization of a touch screen based keyboard. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI'03)*. Miami, Florida, pages 77–84.

Visual data mining: Publications 1–4 in detail

Publications 1–4 relate to creating visual data mining toolboxes. More specifically, Publications 1–3 concentrate on visual data mining (of industrial data) using the *SOM Toolbox* that implements the SOM as a platform for various techniques. Publication 4 introduces *Icasso*, the method for exploratory assessment of results from a randomized algorithm, specialized in assessing results of independent component analysis using visualization. The software uses parts of *SOM Toolbox* as building blocks.

Publication 1 is a review of SOM based tools and methods for industrial data that are developed in the *Intelligent Data Engineering Research Group* at Helsinki University of Technology. The review concentrates on multisensor time series data originating from process industry. It pursues a typical data mining setting for industrial data and presents the benefits of the SOM to the exploratory data analysis process. Four main themes are considered: data survey, time series visualization, correlation and outlier visualization, and a research software, the *SOM Toolbox*, that collects the basic SOM and the tools developed during the research for public use. The contributions originating from my work are the time series exploring tool—combined with the correlation/clustering visualization presented in Publications 2 and 3, discussing the visualization benefits of the SOM, and participating in planning of the *SOM Toolbox*, especially its visualization framework.

I managed the writing process, set the scope of the paper and made the final edition. The co-authors wrote about their original contributions and participated in writing the introduction and the discussion. The industrial cases are contributed by Dr. Esa Alhoniemi (pulp industry) and Mr. Jussi Ahola (metal industry). These examples highlight how the tools of Publication 2 have successfully been used in surveying industrial data. The usage of the SOM in “correlation hunting” and novelty detection originate from Dr. Juha Vesanto’s and Mr. Jussi Ahola’s work.

In **Publication 2** the SOM is combined with scatterplots using the traditional color linking. The original contribution is to apply color linked scatterplots and the SOM for visualizing (process) data in order to survey local dependencies between a pair of attributes in different parts of the SOM. The method is applied to investigating relations in data deriving from a customer help-desk service. I also demonstrate how cluster structures can be visualized on a SOM by using another proximity preserving projection for the color coding.

In **Publication 3**, the regular grid of SOM visualization is altered toward a presentation where the distances between the map units on visualization approximate the proximities in the data space. I propose using a contraction projection for this purpose. The contraction projection has an interesting relation to the conventional, hierarchical clustering. This type of projection suits well for interactive purposes where the user is able to shift between the clustered and regular display of the SOM.

Publication 4 presents an exploratory method for assessing the results of a randomized algorithm on a data sample by means of exploratory data analysis. More specifically, we present an exploratory visualization method for investigating the relations between estimates from the FastICA algorithm. In our method, the algorithmic and statistical reliability is investigated by running the algorithm many times with different initial values and/or with differently bootstrapped data sets. Resulting estimates are compared by visualizing their cluster structure according to a suitable mutual proximity index. Clusters are used for assessing the reliability of the estimates. Also, more robust estimates can be produced using, e.g., the centrotypes of the clusters. The visual exploration tool provides an additional sanity check on the overall results. The use of the tool is tested and demonstrated on a biomedical data set recorded from the brain using MEG. The results are reasonable in light of known properties of the signals.

The topic and rough solution were proposed by Dr. Aapo Hyvärinen. I selected the methods for clustering and visualization, planned the overall tool and implemented it as a software package *Icasso* utilizing the *SOM Toolbox* and *FastICA for MATLAB* packages. We wrote the paper together.

Data mining and emerging user interface applications: Publications 5–8 in detail

Publications 5–8 are related to applying data mining in mobile context-awareness, personalization and adaptive user interfaces (UIs). Publication 6 might seem somewhat alien in this group since it investigates a rather general problem related to independent component analysis. It is, however, directly related to the methodology used in Publication 5. Publications 5 and 7 take some steps in incorporating data mining methods into context-awareness. As explained before, we take an exploratory approach: we investigate and demonstrate feasibility of principal component analysis, time series segmentation and independent component analysis in finding meaningful patterns in data that describes the user context. Ultimately, the aim would be to make a mobile terminal capable of sensing a user's context and usage patterns and learning to adapt to those favorably. Publication 8 describes the ergonomically personalized adaptive keyboard, e.g., for a mobile device. The application represents an example of gaining new ideas through the visualization of data mining algorithms. Here, the common visual demonstration of a vector quantization algorithm in two dimensions is turned into an adaptive keyboard.

Publication 5 introduces the context data and the mobile context analysis problem that are further used in Publication 7. The paper reports results on preliminary exploration on the data using principal component analysis (PCA) and independent component analysis (ICA). The results show a clear clustering structure of the context data visualized in different ways. As a result, we were able to demonstrate that the information contained in the original could be symbolized and transformed into higher-level context by rather simple statistical methods. On the other hand, the results suggested that linear, instantaneous ICA could not extract very interesting components from the data. It became evident that a temporal analysis, i.e., time series analysis would be needed in order to proceed on the problem and this is pursued in Publication 7.

The co-authors were responsible for data collection, feature extraction from the original signals, i.e., the "context atom" creation. I composed the study on the context atoms together with Dr. Mäntyjärvi, and I was responsible for implementing the experiments and visualizations using PCA and ICA. We wrote the paper and discussed the results together.

Publication 6 describes a specific non-linear mixing model for independent binary sources. In this non-linear model, the summation of the linear mixing is replaced by Boolean operator OR and the multiplication by AND. The model might be applicable in the problem described in Publication 5 if one assumes that the aforementioned "context atoms" are binary and generated from independent binary sources (contexts). The same data model is encountered also, e.g., in document analysis. Instead of using any model-specific algorithm, Publication 6 studies experimentally how a heuristic based on the linear ICA performs on this data model. More specifically, we first apply the FastICA algorithm and then scale and threshold the linear mixing matrix in order to form the basis vectors of the binary model. It turns out that this method works reasonably well for signals that are sparse enough. In general, for this type of problem, skewness as a basis for the objective function works better than kurtosis, but interestingly, if the signals are corrupted by additional noise, kurtosis gives better results.

I suggested the topic and the heuristics, designed and conducted the experiments while Dr. Hyvärinen guided the work and discussed the results. We wrote the paper together.

In **Publication 7** we apply time series segmentation into similar context data that was used in Publication 5. It seems that the resulting segmentations are in accordance with the real world situation even when the segmentation criteria is as simple as the within-segment variance.

Publication 7 has also an algorithmic contribution: We point out that dynamic programming, which produces an optimal segmentation, is computationally quite intensive for mobile computing. Some computationally less intensive but suboptimal solutions for the problem are known, e.g., the traditional top-down binary split or merge-split type algorithms. We propose and analyze two presumably novel variants of the merge-split optimization method. These randomized and greedy algorithms are called local iterative replacement (LIR) and global iterative replacement (GIR). The quality of the results produced by GIR on the context data set compare favorably to the binary-split algorithm.

I elaborated Prof. Heikki Mannila's original idea of LIR with co-authors. I guided making the experiments and was responsible for discussing the application to the context data. Ms. Johanna Tikanmäki and Mr. Kalle Korpiaho implemented the algorithms and made the performance tests, and also contributed to elaborating the randomized algorithms. I bear the principal responsibility for writing the paper.

Publication 8 continues the research on adaptive UI in a mobile phone but on a different aspect: the physical appearance of the UI. The decreasing size and the expectations for personalization challenge the conventional, rigid keypad design.

We present an idea of an adaptive soft keyboard where the aim is to obtain an ergonomically better, more comfortable keyboard. The method starts from some conventional keypad layout, but it gradually shifts the keys into new positions according to the user's grasp and typing pattern. The purpose is not, however, to change the original global order of the keys but just adjust the positions of the keys. Thus, our application avoids a common pitfall of adaptive UIs: changing the conventional habit drastically that is known to rather distract than help the user. We describe a simple implementation on a numeric keypad and report experiments on a small group of users.

I elaborated the idea with co-inventor Dr. Jani Mäntyjärvi who contributed especially to the overall system design. I implemented the software that was used for realizing the system, performing the user tests and analyzing the results. Ms. Jonna Häkkilä and Mr. Petri Kangas, shared the principal responsibility for the usability test design, discussion on human-computer interaction, the testing itself and the hardware setting. Ms. Häkkilä and I shared the main responsibility for writing the paper.

1.2 Outline of the introductory part

This chapter of the introductory part has expressed the aims of the thesis and scientific contributions of the publications. The rest of the introductory part is divided into chapters as follows:

Chapter 2 is an overview of the research fields that the methods in this thesis relate to: data mining and knowledge discovery, information visualization, and visual data mining. **Chapter 3** provides a short tutorial on the data mining and visualization methods appearing in this thesis and motivates their use. Process industry and brain imaging are not within the scope of this thesis as such. Therefore, in Chapter 3, I concentrate just in the (visual) data mining methods that can be utilized in these fields. Meanwhile, emerging UI paradigms, like context-aware UIs, do form a central part of the thesis. **Chapter 4** is dedicated for a literature review on applying machine learning to context-awareness, personalization, and adaptive UIs. Finally, **Chapter 5** contains a short conclusion and some thoughts about further research.

On one hand, the introductory part of this thesis serves as a general literature review. On the other hand, it presents some details that have been only referenced in the original publications. The relation of the introductory part to the original publications is commentary. That is, most of the technical substance of the publications is not reproduced here.

Chapter 2

Data mining and visualization

2.1 Data mining

2.1.1 What is data mining?

Data is a Latin word having roughly the meaning "things that have been given". For us, it refers to a collection of numbers or other symbols which have been given in a form that can be processed by a computer. It might originate from scientific experiments, industrial processes, corporate databases, or collections of digitized documents. Such collections of data, as mass like entities, are of little value unless they can be transferred into *knowledge*: familiarity, awareness, or understanding gained through experience or study.

The computerized process that combines methods for studying large amounts of data is known as *knowledge discovery (in databases)* (KDD), a research field that started in the late 1980's (Cios et al., 1998) and has gained much interest for quite obvious reasons. The World Wide Web (WWW) gives a good—but just one—example why this is the case. KDD is often a complex and iterative process rather than a linear process from data to knowledge. The process involves (Fayyad et al., 1996)

- gathering domain specific knowledge, managing the data base, data preparation and data quality issues (Kim et al., 2003; Pyle, 1999),
- extracting the relevant features from the data,
- data mining: finding the interesting patterns, rules or models (Hand et al., 2001; Kleinberg et al., 1998),
- assessing the results, and
- exploiting the knowledge in research or in business.

There is a notable endeavor to standardize the KDD process in Cross-Industry Standard Process for Data Mining (CRISP-DM model) (Shearer, 2000).[1]

Data mining can be seen as the specific part of KDD where informative and novel patterns, rules, or models are discovered (Cios et al., 1998; Hand et al., 2001). The patterns, rules, or other descriptions should be interesting and useful to the data owner, and be something previously unknown (Fayyad et al., 1996; Hand et al., 2001). "Interestingness" of a pattern can be measured with utility,

[1] at `http://www.crisp-dm.org` [referred 14 Apr 2004].

i.e., the ability to suggest concrete, profitable actions (Kleinberg et al., 1998). Sometimes, terms "KDD" and "data mining" are used interchangeably, e.g., (Shearer, 2000; Vesanto, 2002), and we will not make a strict difference between these two concepts in this work.

Data mining resides in the junction of traditional statistics and computer science. As distinct from statistics, data mining is more about searching for hypotheses in data that happens to be available instead of verifying research hypotheses by collecting data from designed experiments (Glymour et al., 1997). Data mining is also characterized as being oriented toward problems with a large number of variables and/or samples that makes scaling up algorithms important. This means developing algorithms with low computational complexity, using parallel computing, partitioning the data into subsets, or finding effective ways to use relational data bases (Kepner and Kim, 2003; Lawrence et al., 1999; Provost and Kolluri, 1999). The process- and utility-centered thinking in data mining and knowledge discovery is manifested also in the reported, commercial systems, e.g., (Hsu et al., 2002).

2.1.2 Data mining tasks

Hand et al. (2001) recognize some characteristic tasks that are encountered within data mining research: predictive modeling, descriptive modeling, discovering rules and patterns, exploratory data analysis, and retrieval by content.

Predictive modeling includes many typical tasks of machine learning such as classification and regression. In this thesis, the emphasis is more on *descriptive modeling* that is ultimately about modeling all of the data, e.g., estimating its probability distribution. Finding a clustering, segmentation or informative linear representation are common subtasks of descriptive modeling, and their applications are important in this thesis.

Particular methods for *discovering rules and patterns* emphasize finding interesting local characteristics, patterns, instead of global models. Examples—in the context of the methods and problems encountered in this thesis—include finding local patterns in time series (Oates, 2002; Patel et al., 2002) or atypical items (outliers) using the self-organizing map (SOM; Sec. 3.2.12) (Muñoz and Muruzábal, 1998).

Different methods of *machine learning* are used in data mining to perform these tasks. In "learning from data" (Cherkassky and Mulier, 1998), "learning" is a metaphor that is used to describe finding the interesting patterns. "Learning" does not refer here to the analyst but to a learning machine, a system that is capable of extending predictive power outside the sample of data and finding generalizations or reductions of the data. A list of typical learning tasks—density estimation, classification (pattern recognition), regression, and clustering (Cherkassky and Mulier, 1998)—echo some of the data mining tasks.

The task of *exploratory data analysis* is most important in this thesis. The name originates from the seminal work of Tukey (1977) and refers to an approach in statistics where the focus is on creating hypothesis instead of rigorous statistical inference for testing them (Glymour et al., 1997; Tukey, 1977). Statistical graphics assist in this task (Cleveland and McGill, 1987; Cleveland, 1994; Tukey, 1977). At the stage of exploratory data analysis, the task is not yet to confirm the validity of the results but to provide clues for further research.

Finally, the task of *retrieval by content* has many popular applications including queries from WWW and other large collections of data. For example, of the references in this thesis (Honkela et al., 1996; Kohonen et al., 2000; Laaksonen et al., 2000; Pampalk et al., 2002a; Skupin, 2002; Yang et al., 2003) relate to this task.

2.1.3 Related research fields

Data mining benefits of computational methods from various research fields of machine learning. Consequently, many features of this thesis have appeared elsewhere than in conferences on data mining in particular.

Publication 1 has appeared in a book devoted to *pattern recognition* which is the traditional field that considers many problems important in data mining. Typically, the aim in pattern recognition is to classify some sensed patterns in predefined classes. Automated handwritten character recognition in mobile equipment gives an illustrative example (Vuori et al., 2000). However, pattern recognition is not confined to the learning task of classification. At least, the task of clustering (unsupervised classification) is usually considered in pattern recognition textbooks, e.g., (Duda et al., 2001; Schalkoff, 1992).

Algorithms for machine learning are studied also under the subject of *artificial neural networks* (ANN) that has gained inspiration from the information processing in real nervous systems, see e.g., (Flanagan, 2001; Kohonen, 2001). Characteristic benefits of ANN algorithms are that they adapt (learn) on-line, are noise (fault) tolerant and often support parallel implementation (Kohonen, 2001). Of the methods used in this thesis, the SOM (Kohonen, 1982) has a special position in ANN theory. Also, learning algorithms related to principal component analysis (PCA; Sec. 3.2.4) (Oja, 1982) and independent component analysis (ICA; Sec. 3.3.1) (Jutten and Herault, 1991) have been influential.

In practice, many ANN conferences and journals are forums for machine learning in general. For example, Publications 3 and 4 have appeared in ANN conferences. The substance of this thesis is, though, more about exploratory data analysis than about the paradigm of ANNs. Nevertheless, the on-line performance and adaptivity are often of utmost importance, e.g., in the context-aware applications (Flanagan et al., 2002).

2.1.4 Visualizing results of data mining

Results produced by data mining methods can be assessed by statistical means or codified using rules. For example, in the context of data mining using the SOM, one can check whether significant differences exist between the clusters, or derive rules that describe the clusters (Vesanto and Hollmén, 2002). In this thesis, instead of an analytical approach, we usually concentrate on the visualization of the results by linking them into some other representation of the original data. In Publications 1 and 2, the structure of process data detected by a SOM is linked with color coding and manual brushing to the original process time series and scatterplots between interesting variables. In Publication 7, the results of time series segmentation are related to a video stream which gives an automated narrative, a series of images of what happened in the putative, interesting context changes. The aim of such tools is to assist in the exploratory phase of the research.

In the following sections, we will visit visual data mining, the research area between visualization and data mining, as well as visualization in general.

2.2 Visualization

2.2.1 What is visualization?

External representations for problem solving and communication

The word *visualization* has a meaning of “picturing something in one’s mind”, i.e., imagining. But it has also a reversed meaning, making pictorial *external representations for problem solving* (Zhang, 1997) and for communication. Visualizing ideas by drawing and sketching is kind of “thinking with pictures", and it has evolved hand in hand with engineering practices (Ferguson, 1977). Sketching pictures makes it possible to design complex systems and mediate tacit, informal knowledge within engineering communities and organizations (Henderson, 1999).

External representations provide means of perceptual problem solving by redirecting cognitive operations (Zhang, 1997). A task might be completely different depending on the form of presentation. Sometimes, the solution can be directly perceived and does not need to be interpreted explicitly in high-level cognitive operations. For example, a time series of measurements can be plotted as a curve, a line graph, to make certain patterns immediately perceivable.

Visualization transforms cognitive work on symbols into processing of visual features that may in some cases be preattentive (Healey et al., 1996), which allows fast parallel processing of the features.[2] One can say that the visualization "amplifies cognition", and the several ways in which this happen are discussed also in (Card et al., 1999a) with further references. A quite obvious detail is that visualizations decrease working memory load while performing cognitive tasks (Zhang, 1997).

Zhang (1996) points out three tasks of data graphics:

- information retrieval (value lookup),
- value comparison among one attribute or between attributes, and
- finding emergent patterns in data.

All functions appear already in some most typical examples of data graphics: *line graphs, scatterplots* and *scatterplot matrices* (Cleveland and McGill, 1984; Cleveland, 1994). In a line graph of a time series, we can both think of finding a value for a certain time instant, or comparing values—but we can also see the slope of a curve or other patterns such as cycles or trends. While an alphanumeric table can be more accurate in providing exact data values, a line graph illustrates how a visual form immediately provides access into patterns in data that are awkward to find in a table (Lohse, 1997).

Different categories and aims of visualization

Card et al. (1999a) define the concept of *information visualization* as "the use of computer-supported, interactive, visual representations of abstract data to amplify cognition".[3] This distinguishes the archetypal information visualization from archetypal *scientific visualization*, e.g., (Nielson et al., 1997). In scientific visualization, the objects that are shown have typically quite a clear physical meaning; visualization of flow and vortex, geological/geophysical, molecular structure or medical imaging are a few examples of the application fields. A conventional, static diagram on paper or screen is, according to the taxonomy of Card et al. (1999a), called *data graphics*. Expositions of different data graphics, scientific and information visualization techniques in general can be found in (Bertin, 1983; Card et al., 1999b; Cleveland, 1994; Fayyad et al., 2002; Nielson et al., 1997; Tufte, 1983, 1990), and introductions to multidimensional abstract data visualization in (Card et al., 1999a; Grinstein and Ward, 2002; Hoffman and Grinstein, 2002; Keim, 2002; Wong and Bergeron, 1997).

We are interested especially in the *exploratory* purpose of visualization for discovering new results. Another familiar purpose of visualization is to be *presentational* in order to illustrate known results in papers, books or lectures (Kosslyn, 1994). That is, visualizations are also means of *communication*. A data graph may even be a part of aesthetic line in books, magazines, advertisements, or corporate handouts. The *decorative* use of data graphics can create "disinformation" when done

[2] A task that can be performed in a multielement display in less than 200 ms is considered preattentive (Healey et al., 1996).

[3] Card et al. (1999a) also define that *visualization*, in general, refers to "interactive use of computer to visualize relations in data". In this thesis, the word "visualization" is used in a more everyday sense. It may refer to any pictorial representation of data or concepts.

inexperienced (Tufte, 1983). On the other hand, skillful design of misleading visualizations can be deliberately used in propaganda.

Craft and tradition of visualization

The works of J. Bertin and E. R. Tufte, e.g., (Bertin, 1983; Tufte, 1983, 1990) are recognized as seminal in the area of information visualization in many surveys (Lohse, 1997; Card et al., 1999a; Ware, 2000). These works are not focused on computerized visualizations but are important sources of profound graphical experience, tradition and best practices.

Tufte crystallizes making good visual representations in conjectures of graphical excellence and a couple of quantitative measures. The graphical excellence is about "communicating complex ideas with clarity, precision and efficiency". The aim to present the original values of data with precision (i.e., lookup and comparison tasks). Tufte encourages using minimalistic and subtle design (within reason) to convey the necessary information. For example,"data-ink ratio" should be high meaning that as little "ink", i.e., intensity and area, should be used for objects that are "not data", e.g., axes or tick marks. Tufte's works are also about the history and origins of data graphics and present a lot of interesting examples of both successful and disinformative visualizations.

While Tufte present an approach based on general graphical design and aesthetic guidelines, Bertin (1983) describes a detailed theory and taxonomy of data graphics. Bertin's theory is based on semiology: He treats maps and data graphics as a language where the elements of visualizations, signs, interact. Bertin also classifies the graphical attributes, e.g. luminance, color, texture, position, and size of an object, in a way that presents the most suitable purpose of use.

Toward science of information visualization

Bertin's theory is a rather subjective one, but it contains many of the same elements as later work in the field that is explicitly based on psycho-visual and cognitive research results (Card et al., 1999a; Mackinlay, 1986; Zhang, 1996). Such results give a measurable, *scientific* background for information visualization. They tell how the elements and attributes of visualization can take advantage of universal, low-level sensory mechanisms. Card et al. (1999a) and Ware (2000) survey such results and their application to information visualization.

What has been found during visualization research can be expressed as general design principles or specific rule-of-thumbs included in reference books, e.g., (Cleveland, 1994; Eick, 1997; Kosslyn, 1994; Ware, 2000). However, designing a visualization takes a lot of effort and it is not guaranteed that the designer's graphical skills are up to the best level. Thus, automated design would be beneficial. For example, Mackinlay (1986) presents a cognitive model of graphical perception and interaction and a methodology, *A Presentation Tool*. Lohse (1997) surveys further models in automating the evaluation and design of data graphics. However, according to de Oliviera and Levkowitz (2003), the cognitive models are still restricted to a set of basic data graphic tasks, and are not sufficient for the growing number of sophisticated exploratory and interactive visualization techniques. Eventually, a visualization system may have some specific features that are "perceptually tuned". For example, related to Publication 3, Kaski et al. (1999) present a method for color-coding a cluster visualization of the SOM that is based on using a perceptually uniform colorspace.

Empirical user studies are important in the research of, often complex, human-computer interaction systems. This applies to visualization systems and techniques as well. In information retrieval, there are a number of diverse studies: Chen and Yu (2000) survey and compare a selection of these. For exploratory visualization methods of abstract data, such studies are harder to find. Grinstein et al. (2002) and Keim and Kriegel (1996) provide formative examples. They compare the performance of certain visualization methods on benchmarking data sets with known clusters, outliers, or rules.

While these two studies interestingly highlight differences in applicability of visualization techniques on different type of data and tasks, a potential weakness is that the evaluation is based on the personal opinion of a small group of experts.

Understanding novel visualization methods requires expertise. Consequently, it is hardly reasonable to expect that all characteristics of an advanced information visualization method would be immediate for an unexperienced analyst. This also lays challenges for fair evaluation of the visualization systems and methods (Keim and Kriegel, 1996). For example, in a study on information retrieval tasks (Sebrechts et al., 1999) a difference between “computer skilled” and “layman” was apparent when using a 3D visualization system. Even, in traditional data graphics, learning has been shown to improve reading and understanding of the graphics (Lohse, 1997). Undoubtedly, many aspects of visualizations are conventions that have to be learned (Ware, 2000).

2.2.2 Visual data mining

What is visual data mining?

This thesis does not contribute to the research of the psycho-visual basis of visualization, but it focuses on the “craft” or ”engineering” side of visualization and uses methods which

- are intended for multidimensional (abstract) data visualizations,
- are implemented in toolboxes using a computational environment that supports interactive use and graphics, and
- combine the data mining approach and visualizations.

This kind of combination is studied in particular in *visual data mining* (VDM) (Keim, 2002; de Oliviera and Levkowitz, 2003; Wong, 1999). In general, it can be said that VDM lies in the junction of data mining and visualization. In addition to being data mining by *visual data exploration*, VDM can also aim at visualizing the results and operation of computational data mining methods. This should make the process of data mining more transparent to the analyst. A tightly coupled VDM system supports using visual abstractions for gaining insight into the data and operation of algorithms, instead of just conveying results of automated decisions (Wong, 1999; de Oliviera and Levkowitz, 2003). Furthermore, a VDM system can guide the user through a data mining process and be coupled with decision making, e.g., for substituting analytical decisions by human expertise when needed.

As data mining, VDM and related fields collect a large variety of methods; de Oliviera and Levkowitz (2003) survey the plethora of techniques, methods, software, and terminology. They also compare some attempts to categorize and formalize VDM and information visualization methods (Card et al., 1999a; Keim, 2002; Chi, 2000). Keim (2002) uses three main aspects for classifying VDM methods:

1. The *nature of the data or object(s) to be visualized* where Keim (2002) differentiates 1D (time series) data; 2D–3D data, multidimensional data, data consisting of text or document collections, data that is mainly about hierarchies or graphs, and algorithms/software.

2. Characteristics of the base *visualization technique*. For example, there are “standard techniques” such as scatterplots, bar or pie charts, and “alternative” axis techniques, e.g., the parallel coordinates method (Sec. 2.2.3). Furthermore, Keim (2002) distinguishes between using hierarchical division of space, graphs, glyphs (Sec. 2.2.3), and dense pixel methods. Dense pixel methods aim at a very high data density on the visualization (Keim and Kriegel, 1996;

Keim et al., 2002). Obviously, the nature of the data has an impact on the proper selection of the method (Card et al., 1999a).

3. The *interaction and distortion technique* used for interactively zooming, filtering, browsing, querying, linking, brushing, or distorting the visualizations (Sec. 2.2.3).

2.2.3 Some basic building blocks of visualizations

Differentiating the computational method and visualization

The visualization techniques are processes that can be broken down into several stages where the nature of the data and operations are different. Chi (2000) breaks down visualization techniques into several stages and operators, and proposes a taxonomy of visualization processes that is useful also to software designers. We will not pursue this particular process model in detail but point out a general remark: Casually, we say, e.g., that the SOM or PCA are used for visualizing data. Actually, the process consists of different phases. First, one transforms the data into features using a computational method (such as SOM or PCA). Second, some attributes of the results are *visualized*, i.e., mapped into a *graphical attribute* such as position. The mapping can be done in several ways and, consequently, the visual appearances can be quite varied.

Graphical attributes and dimensional anchors

Graphical attributes include, e.g., texture, color, size, shape, orientation, and intensity (gray scale) of a marker or an area.[4] Other graphical attributes than position are sometimes called *retinal attributes* (Bertin, 1983; Card et al., 1999a). Our average capability of accurate recognition (absolute judgment) of different levels of magnitudes in visual attributes is not very high. Furthermore, different contrast and illusory effects emerge in case of many visualizations and may change the correspondence between the original and perceived value (Tufte, 1983; Ware, 2000).

However, we can detect astonishingly small relative differences among graphical attributes. Consequently, the idea of visualization is that graphical markers are not independent, but are in correspondence with each other (Bertin, 1983). Visualizations typically use some frame of reference for the graphical markers. One way to describe such references is the concept of *dimensional anchors* (Hoffman et al., 1999). A typical example is a scatterplot where the location of the markers is “anchored” to the scales along two perpendicular axes that may also include a grid which helps determining the values. The framework of dimensional anchor extends to other presentations, e.g., to *parallel coordinates*, a method that is meant especially for displaying multidimensional data (Inselberg and Dimsdale, 1990). See Fig. 2.3 in Sec. 2.2.5.

In general, specially manipulated axes (e.g., stacked or hierarchical axes) and other types of dimensional anchors give rise to sophisticated visualization methods reviewed in (Card et al., 1999b; Fayyad et al., 2002; Keim, 2002; Nielson et al., 1997; de Oliviera and Levkowitz, 2003).

Movement and stereoscopic vision bring in further options. Both in scientific and information visualization one can use immersive, virtual and animated scenes using the realm of techniques in computer graphics (Haase et al., 1997). However, in this thesis, our scope is limited to visualizations that remain in two dimensions and do not use animation.

[4] Somewhat the same graphical attributes are encountered in computer graphic systems (Hearn and Baker, 1997), and Cleveland and McGill (1987) note that the graphical attributes need not to be the most fundamental in perceptional theory since their aim is to assist the graph maker to manipulate graphs conveniently.

Examples of using graphical attributes

Different graphical attributes have different resolution and other characteristics. For example, the function between the physical and perceived value of an graphical attribute varies (Ware, 2000), and different graphical attributes imply different types of data (Zhang, 1996). Not surprisingly, *position* (on a scale) and line length are the attributes that are often referred to be the best choice for quantitative coding on a ratio scale (Card et al., 1999a; Cleveland and McGill, 1987; Lohse, 1997; Zhang, 1996).

While *gray level* and *size* are somewhat suitable for quantitative coding, they suffer from contrast effects and are less accurate in general. They are often recommended for coding a few ordinal variables in value lookup and comparison (Card et al., 1999a; Ware, 2000; Zhang, 1996). But misinterpretations can happen also when a “too effective” graphical attribute is used. For example, if one codes a nominal variable using a position on a scale this implies ordering that is not present and may lead into false impression on the data (Mackinlay, 1986; Zhang, 1996).

Color is usually recommended especially for nominal (categorical) labeling of markers (Card et al., 1999a; Zhang, 1996). According to Healey and Enns (1999), around seven carefully selected colors can be distinguished rapidly and accurately. However, the use of color in directly conveying dimensional information, e.g., in cluster visualization is not completely ruled out (Ware and Beatty, 1988).

Clearly identifiable *shapes* are also often used for class labeling, e.g., in scatterplots. See Fig. 2.1 (Sec. 2.2.5). On the other hand, one can also parametrize different attributes of a shape, say width and length. This technique is used in *glyph* (or icon) visualizations, e.g., (Chernoff, 1973; van Walsum et al., 1996). See Fig. 2.2 (Sec. 2.2.5). A glyph can be a composite of many different adjustable shapes, lines, and colors: Chernoff (1973) presents a famous glyph visualization that uses stylized human faces to represent multidimensional data.

Shading of an area using different gray levels or color codes is often used to show a function on 2D plane, e.g., in U-matrices or component planes (Sec. 3.2.13). A carefully designed color scale can reduce the contrast effects and be better for value lookup tasks on a 2D surface plot (Ware, 1988). On the other hand, a color scale may cause artificial contours giving raise to false interpretations about general patterns (Ware, 1988, 2000). Now, a choice of preference comes in: is the aim to assist seeing general patterns or value lookup?

Accurate value lookup—emergent patterns

In general, visual attributes cannot be treated as being independent channels. Some of the attributes form integral pairs, e.g., the hue and saturation of color, or height and width of a shape (Healey et al., 1996; Ware, 2000). Components in an integral pair are not very good for separate value lookup tasks: they form a new, emergent visual pattern. In general, complex combinations of visual attributes cause phenomena that often require detailed study and modeling (Healey et al., 1996; Healey and Enns, 1999; Ware, 1988, 2000). This sets challenges in constructing composite visualizations, e.g., glyph visualizations.

In Publication 5, the slope of the signals is not really of interest, instead the time span of the somewhat discrete events is, so using multiple line graphs might be confusing in this case. A *Gantt chart* is a popular bar chart for showing the interrelationships in projects, schedules, and other time related systems that progress over time. It is quite natural to adopt this way of presenting the data in Fig. 4.2 (Sec. 4.3.3) instead of a line graph. However, in a scatterplot or line graph, detailed recognition of the level of some signal can be seen better. For example, compare the same signal presented as a line graph in Fig. 4.1(a) and a gray level coded bar in Fig. 4.1(b) (Sec. 4.3.3).

Techniques for exploration and interaction

An important part of VDM, visual data exploration, consists of an overview for visually finding interesting subsets, *zooming and filtering* for focusing to interesting subsets, and then *details-on demand* by a drill-down feature or using another visualization technique (Keim, 2002).

A problem that can be solved using interaction is the *focus-context problem* (Card et al., 1999a). For example, if the user zooms into details he may lose the context of the data and in a panoramic view, the details become unreadable. A separate focused view linked to a general view is one solution (Card et al., 1999a). View distortion techniques, like the fisheye views (Leung and Apperley, 1994), can also help in the focus-context problem. Ritter (1999) and Yang et al. (2003) apply different view distortions to SOM visualization.

Brushing (Becker and Cleveland, 1987) and *linking* of plots (Buja et al., 1991) are two common techniques in visualization. Brushing means selecting marks on a view of the data, e.g., a scatterplot, and transferring the selection into other views of the same data. This links the presentations together by using a retinal attribute of the markers. For example, the user can change the color of the interesting data points by brushing them. In Publication 1, a brushing tool is presented for interactively brushing and linking a line graph representation of a time series and a SOM. In Publications 1–3, a color coding is used to link views. Furthermore, views can be linked using explicit connections (like in Fig. 4.1(b)) or movement of the markers (Buja et al., 1991).

2.2.4 Focus on software

VDM is an interactive and iterative process and requires software that should be reconfigurable, general, and widely usable (Wong, 1999). The observations of Pyle (1999) on the importance and time consumption of data preprocessing in KDD process apply also to visual exploration of data. In an expert user study, reformatting and transforming data were evaluated to be at least as important functions as the tasks of exploring and presenting the data (Hibino, 1999).

Consequently, enabling visual data mining require a lot of effort in integrated software systems. Polaris (Stolte et al., 2002) and GGobi (Swayne et al., 2003) are recent examples of visualization systems for data exploration in general. They support various multivariate interactive visualization techniques, provide programming interfaces, specialized formal visualization languages, and are compatible with different databases and data types.

MATLAB and S-PLUS are general-purpose statistical and technical computation environments that also provide a collection of different, interactive visualization primitives (Grinstein and Ward, 2002).[5] Consequently, the *SOM Toolbox* (Vesanto et al., 1999, 2000), briefly described in Publication 1, and *Icasso* in Publication 4 were created using MATLAB since it is used quite commonly in scientific and technical computing. It supports fast prototype creation through an interpreted script language and “toolboxes” for various technical fields such as signal or image processing. Therefore, it is quite beneficial in technical, multisensor data analysis problems. Most visualization examples in this thesis were created using the *SOM Toolbox* that implements various data preprocessing, clustering, and projection methods in addition to the SOM and its visualization.

[5] MATLAB is a registered trademark of The MathWorks, Inc.; S-PLUS is a registered trademark of Insightful Corporation.

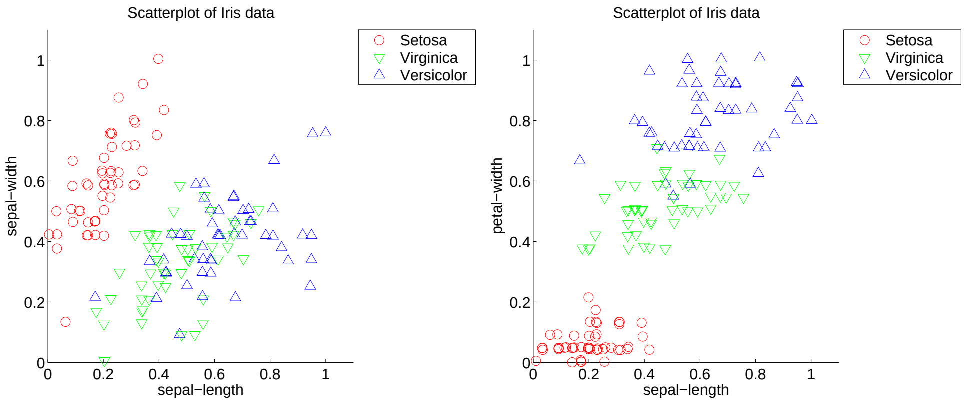
2.2.5 Examples

Iris data

The following examples demonstrate some visualization methods on the famous Iris flower data.[6] The Iris data set is often used in such illustrations of visualization and machine learning. The data contains sepal and petal lengths and widths measured from 150 Iris flowers that belong to three specific types called *Iris setosa*, *virginica* and *versicolor*. Fifty plants stem from each class. The data set is also used throughout the examples in Chapter 3. Each attribute has been linearly normalized (Sec. 3.2.2) between 0 and 1.

Scatterplots

Figure 2.1 shows two scatterplots of two attribute pairs of the Iris data. The class of a data item is shown using shape and color. One can see that the second projection gives somewhat better separation between classes. A small amount of random noise is added into each point since many of them overlap. This technique is called *jittering* (Hoffman and Grinstein, 2002). Many techniques to enhance data analysis using scatterplots, line graphs and other data graphics are described in (Cleveland and McGill, 1984, 1987; Cleveland, 1994).



(a) (b)

Figure 2.1: Two scatterplots of Iris data.

Glyphs and parallel coordinates

Figure 2.2 presents the Iris data as simple glyphs. It is not easy to read any exact values from the glyphs in this view, but they form emergent patterns of shape. The same shape coding is used in Figs. 3.1 and 3.6 in Chapter 3. In Fig. 2.2, the glyphs are organized in reading order—from left to right and top to bottom—according to the rank order of the first principal component (Sec. 3.2.4) of the data.

The glyphs have the same class color labels as in Fig. 2.1. Most people can easily separate the red, green, and blue colors that are the orthogonal primitives of the RGB color space and also fundamental in human vision (Ware, 2000). However, if color separation was really critical, this

[6] documented and available, e.g., in (Blake and Merz, 1998) where attributed to Fisher (1936).

selection would be unfortunate since in the most common forms of “color blindness” differentiating red and green is especially degraded.

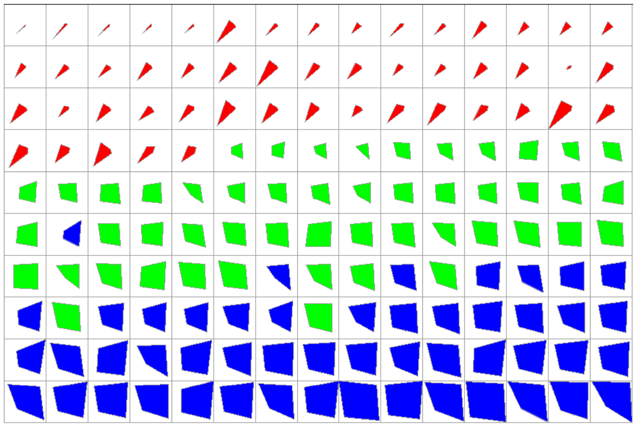


Figure 2.2: Glyph visualization of Iris data items. Each small box contains one glyph. The distances from the center point of the box to the vertices of the glyph codes attributes 1) sepal-length, 2) sepal-width, 3) petal-length, and 4) petal-width counterclockwise starting from the upper left corner. (Red: *setosa*, Green: *virginica*, Blue: *versicolor*).

Another way of visualizing multidimensional data, the parallel coordinate plot is shown in Fig. 2.3. Two clusters are evident also in this plot. In the parallel coordinate plot, one can read the attribute values of a particular item quite accurately, if it is not shadowed by other items. Visual overlapping may be a severe problem in this visualization.

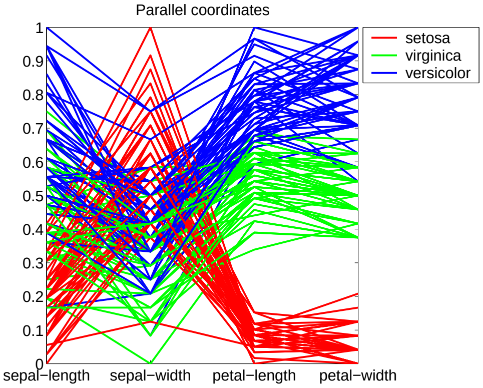


Figure 2.3: Parallel coordinate plot of Iris data.

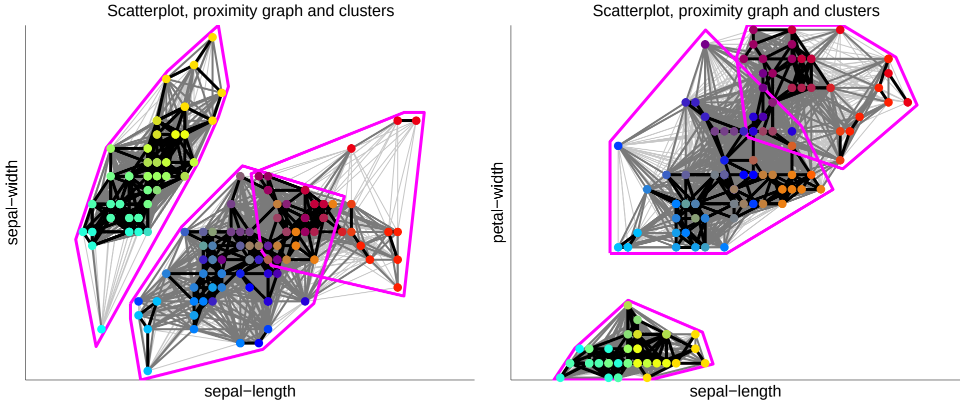
Graphs, containment, linking

Next, we consider linking two scatterplots by a color code, visualizing a graph using explicit connections, and using enclosures to express containment (clusters). The basis of Fig. 2.4 is the same set of scatterplots as in Fig. 2.1.

First, the same kind of graph visualization that is used in Publication 4 is demonstrated. A Euclidean distance between the data items using all four attributes has been computed and the graph line is the darker the shorter the distance is. To be more readable, the graph uses three shades of gray corresponding to three threshold values among distances. Above the third threshold the lines are not drawn. (The exact thresholds are not of interest here.) Drawing all graph lines explicitly is not feasible for large graphs. Abello and Korn (2002) and van Liere and de Leeuw (2003) present more advanced graph visualization techniques. At least, the vertices of the graph can be organized according to the graph topology. Consequently, in Publication 4, a proximity preserving projection in used (Gordon, 1987; van Liere and de Leeuw, 2003). Another way of visualizing distance information is presented in Fig. 3.4 (Sec. 3.2.10) where a clustering method is applied to the data.

The purple convex hulls in Fig. 2.4 are drawn around three clusters (group of data items) that have been selected according to the clustering presented in Fig. 3.4. Evidently, in areas where the convex hulls overlap, it is not clear to which cluster the points belong to.

The color coding of the points is the same as that of the glyphs in Fig. 3.1. The coloring has been given according to projection to the SOM in Fig. 3.5 (Sec. 3.2.13) using the technique described in Publication 2.



(a)

(b)

Figure 2.4: Scatterplots with color linking, proximity graph and cluster visualizations.

Chapter 3

Methods for visual data mining

3.1 Data mining in this thesis

3.1.1 Methods and motivation for their use

Exploratory and descriptive data mining techniques for data description that are relevant here can be divided roughly into three groups:

– *proximity preserving projections* for (visual) investigation of the structure of the data (Sec. 3.2.2–3.2.4 and 3.2.13)

– partitioning the data by *clustering and segmentation* (Sec. 3.2.8–3.2.12)

– *linear projections* for finding interesting linear combinations of the original variables using principal component analysis (PCA; Sec. 3.2.4) and independent component analysis (ICA; Sec. 3.3.1–3.3.2).

Generally speaking, the motivation for using the techniques in most applications of this thesis is to find patterns that reflect different *states or sources in the process* that generated the data. The states are assumed to be reflected in clusters, segments, or linear projections.

What is in common for many algorithms presented in this thesis—the FastICA (Sec. 3.3.2), batch K-means (Sec. 3.2.9), local and global iterative replacement (LIR and GIR; Sec. 3.2.11), and the batch SOM (Sec. 3.2.12)—is that they use *fixed point iteration* of form $A := G(A)$. Especially, in the case of the batch K-means, LIR and GIR, understanding the operation of the fixed point algorithm is very intuitive.

Many theoretical aspects, such as convergence proofs, of the algorithms are left aside. The aim of this chapter is to describe the practical characteristics and operation of the algorithms when this is not done in the supplemental publications.

3.1.2 From data to features

We largely set aside two important preliminaries of a KDD project: the data collection and storage—not to mention the organization, management and effective use of large data bases. We assume that the data is given, suitable for any computation, in a data matrix $\mathbf{X}$ consisting of N vectors $\mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(N)$ that are M-dimensional.

Depending on the context, the elements of the vectors are called *variables*, *features*, *components*, or *attributes*. In this thesis, the attributes are usually real valued. In some cases they are restricted to be between 0 and 1, as in Publications 5 and 7 where some attributes are actually binary (values are 0 or 1). Publication 6 is an exception: there the data and the model is purely binary. However, it is always assumed that the data can be handled with methods for continuous data. Consequently, discrete data types, algorithms and similarity measures are largely beyond the scope of this thesis.

In data mining and pattern recognition, "the data" refers often to the original records, say, in a measurement log file. For example, in Publications 5 and 7, the physical signals from the sensors in a mobile phone form the original, "raw" data. It is practical to transform the original raw data into new attributes that present the data in a way that is known to be relevant in the domain. In pattern recognition literature, this phase of data preprocessing is called *feature extraction*. In Publications 5 and 7, the common line is to extract understandable features, which aims at making it easier to integrate user-defined, rule-based parts to the same system (Sec. 4.3.2). In general, it is customary to select attributes (features) that are the most adequate set for the task at hand; Langley (1994) gives a short review. Since the phase of feature extraction is very domain dependent, it is a major path where the prior, expert knowledge enters the data mining process. Developing suitable feature extraction methods is often a matter of trial and error. Consequently, working toward computer-aided attribute selection and feature extraction in data mining is an interesting challenge; Laine (2003) and Sinkkonen (2003) present two very different approaches related to the feature extraction, weighting, and selection problem, cf. Sec. 3.2.15.

3.1.3 Noise and outliers

The data might include noise or outliers that must be taken care of. In Publication 4, low-pass filtering and dimension reduction using PCA are used to deal with noise and estimation problems. In general, ensuring the data quality and preparing data for the analysis require lots of effort during a data mining project (Pyle, 1999). Outliers—erroneous or drastically different data items— and missing attribute values arise sometimes difficult problems.

Many of the methods considered in this thesis are somewhat sensitive to outliers due to the quadratic error functions they involve. However, we infer that disturbing outliers are gradually pointed out in the course of the exploratory work. Missing values are not a relevant problem with the data sets in this work either.

3.2 Finding patterns by proximity

3.2.1 Visual and automated investigation of data structure

We often wish to make a taxonomy of things based on properties of objects. How could (visual) data mining help in this task? Two dimensional data can be plotted on a scatterplot. In this kind of figure, visual proximity may link together objects into more or less easily identified groups. From this background, a rather general description of *a cluster* seems natural: A cluster is a set of objects which are alike; objects from different clusters are not alike (Jain and Dubes, 1988). In clustering one aims at automatically dividing items into groups by their mutual proximities or other properties. In visual data mining the data is transformed so that it assists visual detection of clusters (among other patterns in data).

When data dimension grows, one needs more advanced techniques than a single scatterplot. For multidimensional data there are specific techniques, such as the parallel coordinate or glyph techniques. In general, one can use other graphical attributes than position; proximity is just one of the *Gestalt grouping principles* that describe qualitatively our tendency to perceive patterns in figures (Koffka, 1935). In addition to proximity these include connectedness; continuity; similarity upon graphical attributes: color, luminance, and shape; symmetry; closures and common fate.[1] Thus, the similarities can be visualized by various means: Empirical results suggest that direct coding of three data dimensions onto RGB *color* coordinates (color of the marker) added to an ordinary 2D scatterplot (position of marker) can be used to detect clusters in 5D data up to some extent (Ware and Beatty, 1988). In typical visualizations of the SOM, detection of clusters is prominently based on visual inspection of *shape* and color (or gray scale) coded patterns and contours: U-matrix, other similar techniques, and color coding of SOM visualization are presented in Sec. 3.2.13. Also, the so called pixel oriented methods (Keim and Kriegel, 1996; Keim et al., 2002) rely on shape and color coding to reveal patterns in data. *Motion* that induces common fate can be used to highlight data structure as well, e.g., the "Grand Tour" method (Asimov, 1985) describes such an approach (Buja et al., 1991).

If data dimension grows high one usually runs out of direct graphical means to present the data—mapping each data attribute onto a separate attribute of the visualization is simply not feasible nor informative. In this thesis, we concentrate on a conventional, computational approach where one tries to reduce the number of attributes and still maintain the information of the original data using proximity preserving projections, PCA, or ICA.

In a proximity preserving projection the focus often changes from presenting the data attribute values to presenting patterns in data. This is because the correspondence between the axes and the data is not straight-forward anymore. In a favorable case, the axes can be interpreted as latent factors related to the phenomena that have produced the data. Whether a physical axis has a meaningful interpretation, depends on the data and the method—as well as the observers familiarity with the method. But in general, the aim of these methods is to bring up patterns, clusters and other interesting structure, from the data.

3.2.2 Proximity and proximity preserving projections

Similarity and dissimilarity

From now on, we use *proximity* as a general notion for measures of *similarity* s_{ij} and *dissimilarity* d_{ij} between data items i and j (Borg and Groenen, 1997). Discussion of different (dis)similarity indices and asymmetric proximities (divergences) that appear in pattern recognition, multidimensional scaling and clustering, can be found, e.g., in (Basseville, 1989; Borg and Groenen, 1997; Everitt, 1993; Gower, 1982; Grabmeier and Rudolph, 2002; Jain and Dubes, 1988). To mention some typical properties (which slightly vary in the literature), we recall that

– all proper (dis)similarities are symmetric,

– it is often required that self-dissimilarities $d_{ii} = 0$ for all objects, all dissimilarities are non-negative (Jain and Dubes, 1988), and self-similarity is the maximum similarity $s_{ii} = s_{max}$ (Grabmeier and Rudolph, 2002),

– while dissimilarity indices often have no upper bound, commonly used similarity indices are typically bounded in $[0,1]$ or $[-1,1]$ (Gower, 1982), and

– metric distances are dissimilarities that further obey the triangular equation $d_{ij} + d_{jk} \geq d_{ik}$ and for which $d(\mathbf{x},\mathbf{y}) = 0$ only if $\mathbf{x} = \mathbf{y}$. It is often possible to construct a simple transformation that changes a non-metric dissimilarity matrix into a distance matrix if necessary (Gower, 1982).

Usually, in data analysis, the proximities between data items have to be computed from the attributes of the items. This opens a fundamentally difficult problem since there is often no explicit model

[1] Motion that visually groups the objects.

in exploratory data analysis, and inevitably, the selection of a proximity measure implicitly introduces profound assumptions of the structure of the data. Selecting a proximity index that makes the exploration successful in clustering or other tasks often calls for a heuristic decision.

Euclidean distance and normalization of attributes

If the data consists of vectors (in $\mathbb{R}^M$), an engineering decision is the everyday notion of proximity, the Euclidean distance. While it is invariant in rotation and translation of the data, it is not invariant for other linear operations, like scaling of the original variables. Consequently, the normalization of the variables becomes of interest. Yet again, doing this favorably for successful exploration is a question that may well call for a heuristic decision.

Normalization to unit variance is a common procedure which Milligan and Cooper (1988) compare to different range normalizations in clustering artificial benchmark data. In this case, the range normalizations performed better. A typical range normalization is to set the minimum value to zero and maximum to one and scale the rest linearly in between. This normalization is used for the data visualization examples in Chapters 2–3. However, it can happen that a specific heuristic hides more rather than brings out patterns, e.g., clusters.

The Euclidean distance is quite ubiquitous in data analysis tools in this thesis. In Publications 1–3, it is used in the *SOM Toolbox* in conjunction with the SOM algorithm. It appears implicitly also in the cost function of the time series segmentation in Publication 7 and very obviously in Publication 8.

Correlation coefficient

In some problems, there exists an intrinsic similarity measure, and the attribute scaling problem can be partly avoided. Comparing independent component estimates in Publication 4 provides an example. There, a natural similarity measure is based on the magnitude of the *linear correlation coefficient* r_{ij} between two attributes i and j:

$$r(\mathbf{X}_{0i}, \mathbf{X}_{0j}) = \frac{\mathbf{X}_{0i}^{T} \mathbf{X}_{0j}}{\|\mathbf{X}_{0i}\| \|\mathbf{X}_{0j}\|} \tag{3.1}$$

where the vectors $\mathbf{X}_{0i}$ and $\mathbf{X}_{0j}$ are attributes i and j, i.e., rows of a centered data matrix $\mathbf{X}_0$. *Centering* means subtracting the average of the data $\mathbf{c} = (1/N) \sum_i \mathbf{x}(i)$ from every data vector in $\mathbf{X}$.

Transformations between similarities and dissimilarities

Sometimes, it is necessary to transform similarities into dissimilarities or vice versa. The correlation coefficient r is limited between -1 and 1, and it can be transformed into a dissimilarity conveniently by $d_{ij} = 1 - s_{ij}$ (Gower, 1982). This transformation is used in Publication 4 for expressing proximities between objects for a clustering tool whose implementation requires a dissimilarity matrix.

In some cases, one has to transform a similarity matrix into a distance matrix $\mathbf{D}$ that *embeds* into a Euclidean space. This means that there is a corresponding point configuration $\mathbf{X}^*$ for which $d_{ij} = \|\mathbf{x}^*(i) - \mathbf{x}^*(j)\|$. A reason for doing the transformation is using a method that assumes Euclidean distances, see, e.g., Nikkilä et al. (2002).

In general, if $\mathbf{S}$ is a positive semidefinite similarity matrix where $s_{ij} \in [-1, 1]$, such as the correlation matrix, then

$$d_{ij} = \sqrt{2 - 2s_{ij}} \tag{3.2}$$

results in a distance matrix that embeds into a Euclidean space (Gower, 1982). In Publication 4, transformation $d_{ij} = \sqrt{2 - 2|r_{ij}|}$ was used since it resulted in a satisfactory visualization. In fact, the resulting dissimilarity matrix does not quite embed into an Euclidean space since the similarity matrix with elements $|r_{ij}|$ is not positive semidefinite in general. It turns out that

$$d_{ij} = \sqrt{2 - 2r_{ij}^2}$$

would give an exact Euclidean embedding since the Hadamard product of two positive semidefinite matrices is positive semidefinite (Horn and Johnson, 1991).[2]

If one needs to transform a dissimilarity into a similarity index, one possibility is an inverse exponential function, e.g., a Gaussian kernel $h(d,\sigma)$ (Grabmeier and Rudolph, 2002) that is used in Publication 3:

$$h(d,\sigma) = \kappa \exp\left(-\frac{d^2}{2\sigma^2}\right) \tag{3.3}$$

where σ is the kernel width and κ a normalization factor.

3.2.3 From distances to a data matrix: principal coordinates

Occasionally, the proximity matrix is given and the corresponding input vectors $\mathbf{X}$ are required. For example, we could know only the mutual road distances between towns and we would like to reconstruct their relative locations (relative coordinates). Given a Euclidean distance matrix $\mathbf{D}$, a corresponding (but possibly rotated) data set $\mathbf{X}^*$ (the coordinates) can be constructed using the method of *principal coordinates* (Gower, 1966; Hand et al., 2001).[3]

When we set $\mathbf{B} = {\mathbf{X}^*}^T\mathbf{X}^*$, the squared Euclidean distances can be expressed as $d_{ij}^2 = b_{ii} + b_{jj} - 2b_{ij}$. Solving matrix $\mathbf{B}$ with elements b_{ij} is presented in detail, e.g., in (Borg and Groenen, 1997). A solution is

$$\mathbf{B} = -\frac{1}{2}\left(\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right)\mathbf{D}_2\left(\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right), \quad \text{where } d_{2ij} = d_{ij}^2, \quad \mathbf{1}^T = \underbrace{[1 \quad 1\ldots1]}_{N}. \tag{3.4}$$

Now, the coordinates $\mathbf{X}^*$ are given by eigenvalue decomposition $\mathbf{B} = \mathbf{E}\mathbf{\Lambda}\mathbf{E}^T$ so that $\mathbf{X}^* = \mathbf{\Lambda}^{\frac{1}{2}}\mathbf{E}$.

The principal coordinates method minimizes

$$\sum_i \sum_j \left(d_{ij}^2 - {d_{ij}^*}^2\right) \tag{3.5}$$

for the mutual Euclidean distance matrix of the data (Mead, 1992). In this thesis, the method of principal coordinates appears in Publication 4.

If the dissimilarities in the principal coordinates method are not Euclidean, the eigendecomposition contains negative eigenvalues and complex eigenvectors. The reconstruction of data using principal coordinates corresponding to the positive eigenvalues may still be satisfactory. This is the case if the discrepancy results from a relatively small perturbation in original, Euclidean distances and leaves the observed dissimilarities to a nearly linear relation to the original ones (Sibson, 1979).

[2] Whether this would have been better for the particular visualization has not been empirically evaluated. It is probable that the effects caused by the reduction of dimensionality and the non-linear projection dominate effects caused by the non-exact embedding, anyway.

[3] also called classical scaling (Mead, 1992; Sibson, 1979) or linear metric multidimensional scaling attributed to Torgerson (1952/1958), e.g., in (Gower, 1966; Mead, 1992).

3.2.4 Principal coordinates and principal components

Principal coordinates are closely related to *principal component analysis* (PCA)[4]. PCA is a classical statistical method for obtaining an orthogonal transformation $\mathbf{E}$ for dimension reduction which maximizes the remaining data variance. The principal components for a data set $\mathbf{X}$ can be computed as follows:

1. Center the data (Sec. 3.2.2).

2. Compute the eigenvalue decomposition $\mathbf{\Sigma} = \mathbf{E}\mathbf{\Lambda}\mathbf{E}^T$ of the sample covariance matrix $\mathbf{\Sigma} = (1/N)\mathbf{X}_0\mathbf{X}_0{}^T$.

3. Arrange the columns of $\mathbf{E}$, i.e., the eigenvectors $\mathbf{e}_j$, and the diagonal elements of $\mathbf{\Lambda}$, i.e, the eigenvalues λ_j, so that $\lambda_1 \geq \lambda_2 \geq \dots \lambda_d$.

4. The eigenvector $\mathbf{e}_j$ gives now the jth principal axis and the component scores $\mathbf{X}^*$on jth principal axis, i.e., the jth principal component $\mathbf{X}^* = \mathbf{e}_j^T\mathbf{X}_0$.

The computational load of a standard implementation of PCA (using eigenvalue decomposition) scales well with sample size but not with the number of dimensions (Hand et al., 2001).

In general, $\lambda_m/\sum_i \lambda_i$ gives the relative amount of variance that each principal component (or principal coordinate) captures. For example, $\mathbf{X}^* = [\mathbf{e}_1 \quad \mathbf{e}_2]^T\mathbf{X}_0$ is an orthogonal projection that contains as much of the original variance as possible in 2D.

If one computes a Euclidean distance matrix for a data sample, the principal coordinates are effectively the same as the principal components of the data set (Gower, 1966). Thus, PCA can also be understood as a proximity preserving projection—since it also minimizes Eq. 3.5 in the case of Euclidean distances. Consequently, in addition to data compression or noise reduction, PCA is traditionally used also for exploratory visualizations Gabriel (1971).

In Publication 5, PCA is used to show the clustering of the state space into few dominant states, contexts, within the data. The new attributes, principal components, can be regarded as low-dimensional, more efficient representations of the same data. In Publication 5, an additional visualization of $\mathbf{E}$ is shown in order to see the relation between the original variables and principal components. This can help in interpreting the meaning of each component. Another way to do this is called a *biplot* (Gabriel, 1971) which can be used to show the projection of the original data coordinate system on principal components.

3.2.5 Whitening using PCA

Principal components can be further normalized to unit variance by

$$\mathbf{Z} = \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{E}^T\mathbf{X}_0, \tag{3.6}$$

and now, any rotated version of $\mathbf{Z}$ is still uncorrelated. This is a way of *whitening* (sphering) the data. Whitening is an essential step in the FastICA algorithm in Sec. 3.3.2.

For Euclidean distances, whitening gives invariance in rotation but in most cases it is not a particularly brilliant way of normalizing data. For example, in clustering it may very well decrease the signal to noise ratio in the data. In many applications, the usual assumption is that the principal components corresponding to small eigenvalues (variance) can be considered as noise.

[4] often attributed to Hotelling (1933) (Hyvärinen et al., 2001; Borg and Groenen, 1997)

3.2.6 Non-linear proximity and topology preserving projections

The restriction with methods of principal coordinates and PCA is that the projection is linear. They also contain a built-in assumption that especially Euclidean distances between the data vectors would be meaningful in finding structure in the data. But in general, even for truly Euclidean distances, a complex data structure might be better revealed and expressed by a method that is designed to follow non-linear structures in the data.

Traditional multidimensional scaling (MDS) methods (Borg and Groenen, 1997; Mead, 1992) aim at preserving distances, or more generally rank distances, in the output faithfully—and they can produce non-linear projections. Typically, MDS methods minimize a cost function (stress) between the original and scaled distances using some iterative optimization algorithm. Also, the cost functions that define Sammon's projection (Sammon, Jr., 1969) and the Curvilinear Component Analysis (CCA) (Demartines and Hérault, 1997) used in this thesis can be presented as

$$\sum_i \sum_{j \neq i} (d^*_{ij} - d_{ij})^2 F \tag{3.7}$$

where F is a function of original or projected distances that characterizes the method.

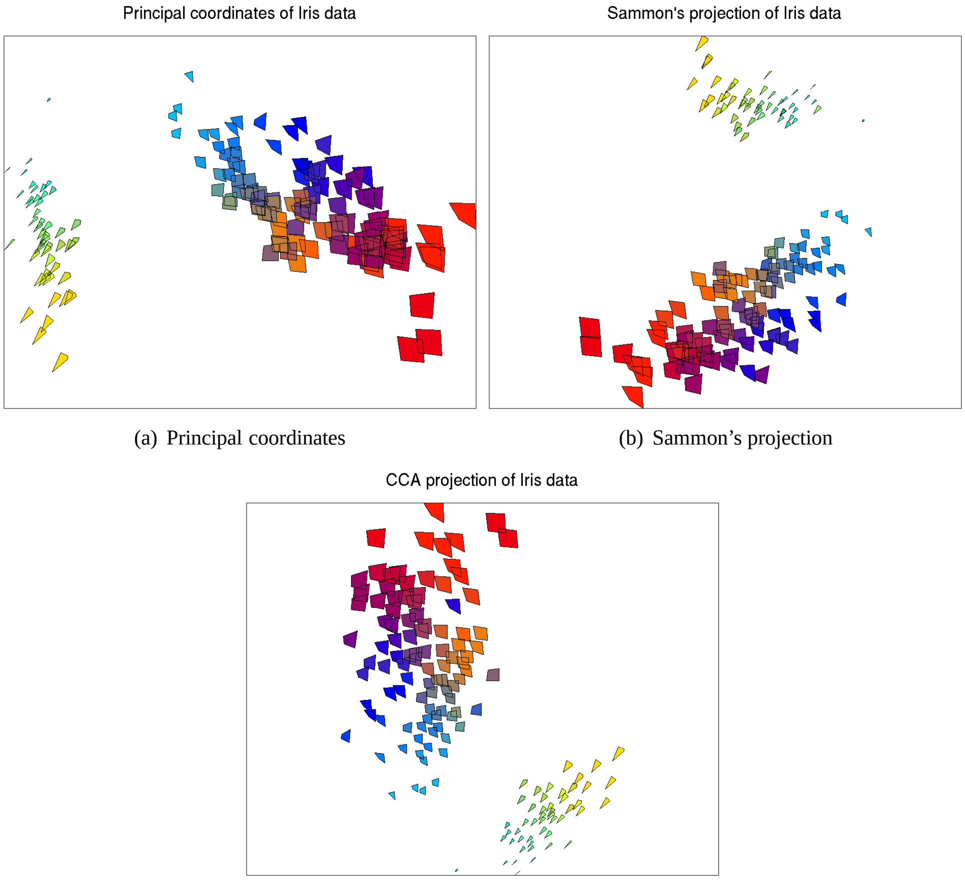
The cost function in Eq. 3.7 with $F = 1$, defines MDS with "raw stress" (Mead, 1992) which aims at global preservation of distances as in the case of principal coordinates method. This is a characteristic of many classic MDS methods (Venna and Kaski, 2001). Function Fcan be used to weight different distances in the projection beneficially:

– In Sammon's projection, Eq. 3.7 has $F = 1/d_{ij}$. Thus, the method gives more weight for original short distances.

– In CCA, F is some bounded, monotonically decreasing function of the projected distances d^*_{ij}, e.g., the Gaussian kernel in Eq. 3.3. The distances that are short in the *projection* have more weight in the stress of CCA. This should favor retaining the local topology of points in the projection (like in a SOM) (Demartines and Hérault, 1997; Vesanto, 2002)—we infer that this should help retaining the projection trustworthiness (see Sec. 3.2.7). A further characteristic of CCA is that the width of F can be decreased along the iteration steps. This makes the target of optimization more complicated than in the case of Sammon's projection.

A method that is also used to reflect the structure of data is the SOM, and in fact, CCA has borrowed many characteristics of it (Demartines and Hérault, 1997)—cf. Sec. 3.2.12. The SOM differs considerably from the classical MDS methods. Although, it aims at retaining the topological relations of the input data, it does not explicitly aim at minimizing a stress, like Eq. 3.7, on distances. Consequently, the SOM is described as a *topology preserving projection*. Furthermore, it also performs a clustering procedure called vector quantization (Sec. 3.2.9). Thus, the SOM is also an example of a *vector quantization-projection method* (Demartines and Hérault, 1993; Vesanto, 1999). Vesanto (2002) discusses further the characteristics of SOM, CCA, and Sammon's mapping and their differences.

Recently proposed proximity/topology preserving techniques include *Isomap* (Tenenbaum et al., 2000) and *locally linear embedding* (Roweis and Saul, 2000) that use the *k* nearest neighbors of data points to reflect the structure of the data manifold in computing the projection. They are claimed to overcome many parameter selection and optimization problems typical for many methods, e.g., for Sammon's projection and CCA.



(a) Principal coordinates

(b) Sammon's projection

(c) CCA

Figure 3.1: Iris data using proximity preserving projections. The data items are shown using the same glyph encoding that in Fig. 2.2 the color of the glyphs comes from the SOM color coding in Fig. 3.5.

3.2.7 Trustworthiness of projections

The obvious graphical method for assessment of the relationship between true proximities and the projected Euclidean distances is a scatterplot between these two (Demartines and Hérault, 1997). Another way is to provide additional visual cues like the weighted graph in Publication 4, to highlight data items that are misplaced by a retinal variable or filter them out (Kaski et al., 2003a). One can also link several projections: Kohonen et al. (1996) advise that a SOM can be visually inspected by using, e.g., Sammon's projection to get a figure of map ordering folding effects. In Publication 2, a color linking coding is used to show relations between a SOM and nonlinear projections.

In general, different MDS and related methods highlight different aspects of data. Consequently, comparing values of their different stress functions is not always sound. Furthermore, stress functions in MDS are not particularly designed to tell how the perceived structure in visualization reflects the structure of data. Considering that proximity[5] usually is a strong indication of similarity or belonging to a group, it sounds reasonable that retaining the local neighborhoods is more important than retaining the global distances in a projection.[6] Based on this idea, Venna and Kaski (2001) have proposed a quantitative measure of projection *trustworthiness* (I_T) for comparing the quality of proximity preserving projection methods. This index is applied in Publication 4; other applications can be found in (Nikkilä et al., 2002; Kaski et al., 2003a).

The trustworthiness index is based on counting “alien items” that are close to wrong (visual) neighborhoods, and therefore degrade the quality of the projection strongly. Consequently, the trustworthiness index is designed to decrease when data items enter wrong neighborhoods in the projection. In addition, the further away a data item jumps into a wrong neighborhood, the more it decreases the quality of projection.

More specifically, let i be some “host” data item, and $\mathrm{rank}_i(j)$ be the rank of the dissimilarity d_{ij} among all dissimilarities from i to the other data items ($j = 1,\ldots,N, \quad i \neq j$). For data item i, the items j with rank distances $1,2,\ldots,k$ are its k *nearest neighbors* (k*-neighborhood*), and we mark this set with $R_{kNN}(i)$. The data items i are projected into, say, 2D space with coordinates $\mathbf{x}^*(i)$. Now, let $R^*_{kNN}(i)$ be the k-neighborhood of data item i *in the projection*, i.e., the items with k shortest distances among $d^*_{ij} = \|\mathbf{x}^*(i) - \mathbf{x}^*(j)\|, \quad j = 1,\ldots,N$. Ideally, neighborhoods $R_{kNN}(i)$ and $R^*_{kNN}(i)$ should remain the same for all data items and neighborhood sizes (i.e., $\forall i,k$).

Now, let $U_k(i)$ be the set of data items which have entered the k-neighborhood of item i in the projection but that are not in k-neighborhood in the original data, i.e, $U_k(i) = \{j | j \in R^*_{kNN}(i) \wedge j \notin R_{kNN}(i)\}$. The degree of dissimilarity between the alien data item and its new surroundings is brought in by using the rank distance between the alien data item and the neighborhood of the host item in the original data space. The trustworthiness becomes then

$$I_T(k) = 1 - \kappa_k \sum_i \sum_{j \in U_k(i)} (\mathrm{rank}_i(j) - k), \quad \kappa(k) = \frac{2}{Nk(2N-3k-1)} \tag{3.8}$$

where κ is to normalize the measure between zero and one for all values of k. This formulation is applicable for neighborhoods $k < N/2$ which should be well enough in practice.

The trustworthiness has a counterpart measuring the *preservation of the original neighborhoods* where the number of points escaping their original k-neighborhood and their rank distance are the basis of the measure.

The trustworthiness of a projection is not a single measure but reflects quality of different neighborhood sizes. In Publication 4, the non-linear projections retain quite well the local neighborhoods that fall into the relatively tight clusters but the actually interesting band in trustworthiness is at neighborhood size that reflects the relations between the nearest clusters.

Figure 3.2 compares the visualization examples of Chapters 2–3 in terms of retaining trustworthiness. Obviously, this is not a comparison of the methods but only the particular figures. No serious attempt in optimization of the methods is made but the *SOM Toolbox* rule-of-thumb heuristics are used. As some of the projections produce ties in distances, the ranking is not unique. Here, the average between the results from most favorable and worst ordering among the ties is presented.

Figures that are produced using the SOM and CCA retain better the local neighborhoods than those made using linear projections, i.e., the principal coordinate plots or the scatterplots of the original attributes. However, on larger neighborhoods Fig. 3.1(a) (based on PCA) has a better trustworthiness index than Figs. 3.6(a) and 3.6(b) based on the SOM. We can also see that the contraction projection of the SOM in Fig. 3.6(b) is comparable with the result of the initial, uniform SOM visu-

[5] in its ordinary meaning of being at a close distance

[6] As a minor note, the larger the distances are the more inaccurate their comparison would be anyway (Cleveland and McGill, 1987).

alization in Fig. 3.6(a) on small neighborhoods but the former retains trustworthiness better on larger neighborhoods. Figure 3.1(c), produced using CCA, seems to be the best in terms of trustworthiness index at any neighborhood size.

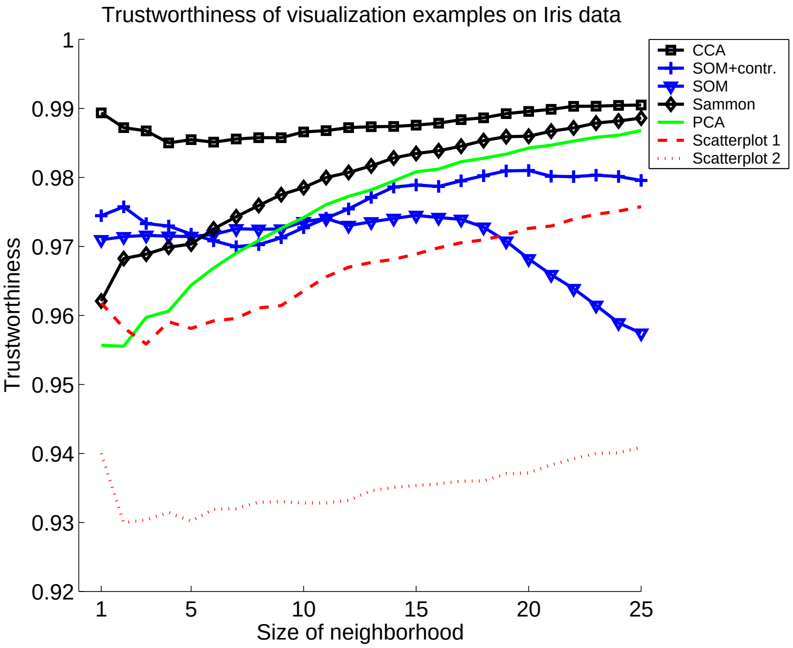


Figure 3.2: Visualizations of Iris data in this thesis evaluated using the trustworthiness index for neighborhoods $k = 1, \dots, 25$. CCA: Fig. 3.1(c), Sammon: Fig. 3.1(b), PCA: Fig 3.1(a), SOM: Fig. 3.5 (Hits) and Fig. 3.6(a), SOM+contraction: Fig. 3.6(b), Scatterplot 1: Fig. 2.1(b), and Scatterplot 2: Fig. 2.1(a).

3.2.8 What is clustering?

So far, we have gone through some methods that can be used to visually detect groups in data. Now, we move on to basic clustering methods that *automatically* divide data items in groups by their mutual proximities. Clustering is indeed a very open problem setting, and there is a considerable amount of specific clustering tasks and clustering methods. Expositions and taxonomies of these can be found, e.g., in (Cormack, 1971; Duda et al., 2001; Everitt, 1993; Grabmeier and Rudolph, 2002; Halkidi et al., 2001; Jain and Dubes, 1988; Jain et al., 1999).

We confine ourselves to dealing mainly with the very basic *hard clustering* where the item either belongs to a cluster or not. Furthermore, the clusters are non-overlapping: a clustering is a partition of the set of all data items $C = \{1, 2, \dots N\}$ into K disjoint clusters $C = \bigcup_{i=1}^{K} C_i$.[7] Some clustering methods construct a model of the input data space that inherently would allow classifying a new sample into some of the determined clusters. K-means and the SOM (Secs. 3.2.9 and 3.2.12) partition

[7] Hard clustering allowing overlapping is called *clumping* (Cormack, 1971), and in *soft clustering*, the data items can belong simultaneously to several clusters with fuzzy or probabilistic memberships, e.g., by generative models (Hand et al., 2001)

the input data space in this manner. Some other methods merely provide a partition of the items in the sample: the agglomerative hierarchical methods provide an example of this case (Sec. 3.2.10).

In addition to the basic cluster analysis tasks, we encounter two examples of clustering methods in this thesis that include *auxiliary constraints*:[8]

– the time series segmentation (Sec. 3.2.11) where the data items have some natural order, e.g., time, which must be taken into account; a segment always consists of a sequence of subsequent samples of the time series, and

– the SOM where the cluster models have prior, fixed topographic relations. This structure is used to capture some of the topographic relations in the data for the purpose of visualization.

3.2.9 Partitional methods

A *partitional* clustering method picks a partition (clustering) which minimizes some cost function. To exemplify, some classical cost functions are given below:

Suggested by the intuitive aim of the basic clustering task, adequate global clustering criteria can be obtained by minimizing/maximizing a function of *within-cluster dispersion* (scatter) $\mathbf{D}_W$, *between-cluster dispersion* $\mathbf{D}_B$, and their sum, the *total dispersion* $\mathbf{D}_T$ that is constant and independent of the clustering. For data in a Euclidean space (Duda et al., 2001)

$$\mathbf{D}_W = \sum_{i=1}^{K} \mathbf{D}_W(i), \quad \mathbf{D}_W(i) = \sum_{j \in C_i} (\mathbf{x}(j) - \mathbf{c}_i)(\mathbf{x}(j) - \mathbf{c}_i)^T \tag{3.9}$$

$$\mathbf{D}_B = \sum_{i=1}^{K} |C_i|(\mathbf{c}_i - \mathbf{c})(\mathbf{c}_i - \mathbf{c})^T \tag{3.10}$$

$$\mathbf{D}_T = \mathbf{D}_W + \mathbf{D}_B = \sum_{i=1}^{N} (\mathbf{x}(j) - \mathbf{c})(\mathbf{x}(j) - \mathbf{c})^T \tag{3.11}$$

where K is the number of clusters, $\mathbf{c}_i$ is the average of the data in cluster C_i, and $\mathbf{c}$ is the average of all data. These quantities can be formulated also for a general dissimilarity matrix (Hand et al., 2001).

The dispersion matrices can be used as a basis for different cost functions. Friedman and Rubin (1967) have proposed two criteria invariant to (non-singular) linear transformations of data based on the dispersion matrices: maximizing $\text{trace}(\mathbf{D}_W^{-1}\mathbf{D}_B)$ and minimizing $\det(\mathbf{D}_W)$. Minimizing the latter gives the maximum likelihood solution for a model where all clusters are assumed to have a Gaussian distribution with the same covariance matrix (Grabmeier and Rudolph, 2002).

The aforementioned criteria may be difficult to optimize (Duda et al., 2001). Therefore, a scale dependent criteria, minimization of $\text{trace}(\mathbf{D}_W)$, has become popular, presumably because it can be (suboptimally) minimized with the fast and computationally light K-means algorithm that is shortly described in more detail. A related, scale invariant criteria is maximizing $\text{trace}(\mathbf{D}_T^{-1}\mathbf{D}_W)$. However, this means effectively the same as first whitening the data and then minimizing $\text{trace}(\mathbf{D}_W)$—with the obvious problems emerging from whitening (Duda et al., 2001).

[8] auxiliary wrt. to the data space.

Vector quantization

Minimization of $\mathrm{trace}(\mathbf{D}_W)$ is the same as minimizing the sum of squared errors (SSE) between a data vector $\mathbf{x}(i)$ and the nearest cluster centroid $\mathbf{c}_j$:

$$\mathrm{SSE} = \sum_{i=1}^{K} \sum_{\mathbf{x}(j)\in C_i} \|\mathbf{x}(j) - \mathbf{c}_i\|^2. \tag{3.12}$$

Eq. 3.12 is encountered in *vector quantization*, see, e.g., (Gersho and Gray, 1992), a form of clustering that is particularly intended for compressing data. In vector quantization, the cluster centroids appearing in Eq. 3.12 are called *codebook vectors*. The codebook vectors partition the input space in nearest neighbor regions V_i. A region V_i is associated with the nearest cluster centroid by $V_i = \{\mathbf{x} : \|\mathbf{x} - \mathbf{c}_i\| \leq \|\mathbf{x} - \mathbf{c}_l\|; \forall l\}$ (*nearest neighbor condition*). Cluster C_i in Eq. 3.12 is now the set of input data points that belong to V_i.

K-means

K-means (MacQueen, 1966) refers to a family of algorithms that appear often in the context of vector quantization. K-means algorithms are tremendously popular in clustering, and often used for exploratory purposes as suggested already by MacQueen (1966). As a clustering model the vector quantizer has an obvious limitation: the nearest neighbor regions are convex, which limits the shape of clusters that can be separated.

We consider only the *batch* K-means algorithm; different sequential procedures are explained, e.g., in (Bottou and Bengio, 1995; MacQueen, 1966).[9] The batch K-means algorithm proceeds by applying alternatively in successive steps the centroid and nearest neighbor conditions that are necessary for optimal vector quantization (Gersho and Gray, 1992):

1. Given a codebook of vectors $\mathbf{c}_i, i = 1, 2, \ldots, K$ associate the data vectors into codebook vectors according to the nearest neighbor condition. Now, each codebook vector has a set of data vectors C_i associated to it.

2. Update the codebook vectors to the centroids of sets C_i according to the *centroid condition*. That is, for all i set $\mathbf{c}_i := (1/|C_i|)\sum_{j\in C_i} \mathbf{x}_j$. See Fig. 3.3.

3. Repeat from step 1 until the codebook vectors $\mathbf{c}_i$ do not change any more.

When the iteration stops, a local minimum for the quantity SSE is achieved (Gersho and Gray, 1992). K-means typically converges very fast. For a discussion on convergence see (Bottou and Bengio, 1995). Furthermore, when $K \ll N$, K-means is computationally far less expensive than the hierarchical agglomerative methods, presented in the next section, since computing KN distances between codebook vectors and the data vectors suffices.

Well known problems with the K-means procedure are that it converges but to a local minimum and is quite sensitive to initial conditions (Peña et al., 1999). A simple initialization is to start the procedure using K randomly picked vectors from the sample as a codebook (MacQueen, 1966), which is compared to other methods in (Peña et al., 1999). A first-aid solution for trying to avoid bad local minima is to repeat K-means a couple of times from different initial conditions. More advanced solutions include using some form of stochastic relaxation (Gersho and Gray, 1992) among other modifications, e.g., (Cheung, 2003; Patané and Russo, 2001).

[9] K-means is known also as the Linde-Buzo-Gray (LBG) (Linde et al., 1980) or the generalized Lloyd algorithm (Gersho and Gray, 1992). MacQueen (1966) attributes the idea of batch K-means to Forgy and Jennifer (1963), independently.

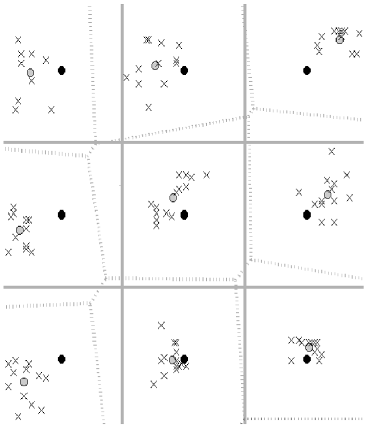


Figure 3.3: One step of K-means. The crosses depict data, the black dots the initial position of codebook vectors, and the gray dots the position of them after performing one step of K-means which in this simple case already reaches the optimum. The solid gray lines are the borders of nearest neighbor regions for the initial codebook vectors and the faint lines for the final codebook vectors. In fact, this figure shows also how the “ergonomically personalized keyboard” (Sec. 4.4.2), described in Publication 8, adapts.

3.2.10 Agglomerative hierarchical methods

The family of partitional methods is often opposed to the *hierarchical* methods. *Agglomerative* hierarchical methods do not aim at minimizing a global criteria for partitioning, but join data items in bigger clusters in a bottom-up manner. In the beginning, all samples are considered to form their own cluster. After this, at $N-1$ steps the pair of clusters having minimal pairwise dissimilarity δ are joined, which reduces the number of remaining clusters by one. The merging is repeated until all data is in one cluster. This gives a set of nested partitions and a tree presentation is quite a natural way of representing the result. See Fig 3.4. In Publication 4, an alternative MDS based visualization (Gordon, 1987) is shown—see also Fig. 2.4.

Gordon (1987) reviews hierarchical clustering in general and different options for cluster merging, or *linkage*, criteria. Here we list the between-cluster dissimilarities δ of some of the most common agglomeration strategies that are also used in Publication 4: the single linkage (SL), complete linkage (CL) and average linkage (AL) criteria:

$$\delta_1 = \delta_{SL} = \min d_{ij}, \quad i \in C_k, j \in C_l \tag{3.13}$$

$$\delta_2 = \delta_{CL} = \max d_{ij}, \quad i \in C_k, j \in C_l \tag{3.14}$$

$$\delta_3 = \delta_{AL} = \frac{1}{|C_k||C_l|} \sum_{i \in C_k} \sum_{j \in C_l} d_{ij} \tag{3.15}$$

where C_k, C_l, $(k \neq l)$ are any two distinct clusters. SL and CL are invariant for monotone transformations of dissimilarity. SL is reported to be noise sensitive but capable of producing elongated or chained clusters while CL and AL tend to produce more spherical clusters (Everitt, 1993).

If similarities are used instead, the merging occurs for maximum pairwise cluster similarity. In Publication 4, Eq. 3.15 is applied after similarity-to-dissimilarity transformation $d_{ij} = 1 - s_{ij}$. This is the equivalent of merging for maximal pairwise similarity $s_{AL} = 1/(|C_k||C_l|)\sum_{i \in C_k} \sum_{j \in C_l} s_{ij}$.

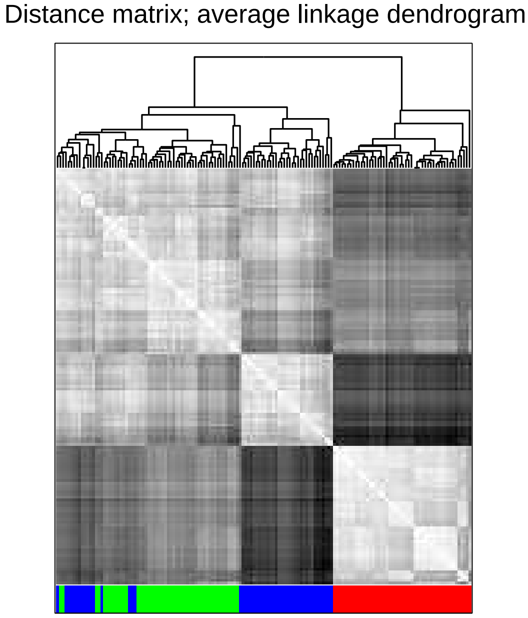


Figure 3.4: Dendrogram produced by AL clustering on Euclidean distances between the vectors of the Iris data set. The data items are at leaves and the height where two branches of the tree join show the linkage distance δ_{AL} where the clusters were merged. The simplest way of achieving K clusters is to cut the dendrogram at a level where K clusters are present. The distance matrix is shown as gray shade representation. Below, the color code shows the true class of the item (Red: *setosa*, Green: *virginica*, Blue: *versicolor*). Note that the order of the leaves is not unique.

3.2.11 Time series segmentation

Problem

A *K-segmentation* divides $\mathbf{X}$ into K segments C_i with $K-1$ segment borders $c_1, \ldots, c_{K-1}$ so that

$$C_1 = [\mathbf{x}(1), \mathbf{x}(2) \ldots, \mathbf{x}(c_1)], \ldots, C_K = [\mathbf{x}(c_{K-1}+1), \mathbf{x}(c_{K-1}+2), \ldots, \mathbf{x}(c_N)].$$

This is the basic time series segmentation task where each segment is considered to emerge from a different model; Gionis and Mannila (2003) address the problem of having less segment models than segments. Furthermore, we consider the case where the data to be segmented is readily available. *On-line segmentation* of a streaming time series is considered in (Keogh et al., 2001).

As in the basic clustering task, we wish to minimize some adequate cost function by selection of the segment border. We stay with costs which are sums of individual segment costs that are not affected by changes in other segments. An example of such a function is an SSE cost function like that of Eq. 3.12 where $\mathbf{c}_i$ is the mean vector of data vectors in segment C_i. There is, of course, a fundamental difference between time series segmentation with SSE cost and vector quantization. In vector quantization, the borders of the nearest neighbors regions V_i are defined by the codebook vectors, whereas in segmentation, the mean vectors $\mathbf{c}_i$ are determined by the segments C_i but cannot directly be used to infer the segment borders.

Minimizing the cost in Eq. 3.12 for segmentation aims at describing each segment by its mean value. It may also be seen as splitting the sequence so that the (biased) sample variance $\sigma^2 = (1/N)\sum_{i=1}^{K}|C_i|\sigma_i^2$ computed by pooling the sample variances of the segments σ_i^2 together is minimal.

Algorithms

The basic segmentation problem can be solved optimally using dynamic programming (Bellman, 1961). The dynamic programming algorithm finds also optimal $1,2,\ldots,K-1$-segmentations while searching for an optimal K-segmentation. The computational complexity of dynamic programming is of order $O(KN^2)$ if the cost of a segmentation can be calculated in linear time. It may be too much when there are large amounts of data. Consequently, greedy algorithms with complexity close to $O(KN)$ have been proposed: Alhoniemi (2003) and Keogh et al. (2001) provide surveys of such segmentations algorithms. Popular algorithms include the obvious hierarchical methods in top-down or in bottom-up manner. Keogh et al. (2001) claim the bottom-up segment formation to be the preferred solution, especially in on-line segmentation.

Another class are the merge-split algorithms of which the local and global iterative replacement algorithms (LIR and GIR) proposed in Publication 7 are examples. They resemble the batch K-means in the sense that at each step they change the descriptors of the partition (segment borders vs. codebook vectors) to match with a necessary condition of local optimum. The LIR gets more easily stuck in bad local minima, and the GIR was considerably better in this sense, yet still sensitive to the initialization. The GIR and LIR algorithms can be seen as variants of the “Pavlidis algorithm” (Pavlidis, 1973) that changes the borders gradually toward a local optimum.

The test procedures in Publication 7 use random initialization for the segments. As in the case of K-means, the initialization matters, and it might be advisable to try an educated guess for initial positions. One possibility to create a more effective segmentation algorithm is to combine several greedy methods. For example, the basic bottom-up and top-down methods can be fine-tuned by merge-split methods. Alhoniemi (2003) compares various combinations of different segmentation algorithms on industrial data. In these experiments, the GIR applied to the initial segmentation obtained from the bottom-up method was the best combination in finding good solutions in reasonable time and computational effort.

Applications

Time series and other similar segmentation problems arise in different applications, e.g., in approximating functions by piecewise linear functions (Bellman, 1961; Cantoni, 1971; Pavlidis, 1973). This might be done for the purpose of simplifying or analyzing contour or boundary lines (Pavlidis, 1980). Another aim, important in information retrieval, is to compress or index voluminous signal data (Shatkay and Zdonik, 1996; Wang and Wang, 2000). Other applications in data analysis span from phoneme segmentation (Prandoni et al., 1997; Xiong et al., 1994) into finding sequences in biological (Bennett, 1996; Gionis and Mannila, 2003) or industrial process data (Alhoniemi, 2003).

3.2.12 Self-organizing map

In this thesis, the self-organizing map (SOM) is used as a data mining method that combines vector quantization and proximity preserving projection. The basic SOM is formed of K *map units* organized on a regular $k \times l$ low-dimensional grid—usually 2D for visualization. See Fig. 3.5.

Associated to each map unit i, there is a

1. *neighborhood kernel* $h(d_{ij}, \sigma(t))$ where the distance d_{ij} is measured from map unit i to others along the grid (output space), and
2. a *codebook vector* $\mathbf{c}_i$ that quantize the data space (input space).

The magnitude of the neighborhood kernel decreases monotonically with the distance d_{ij}. A typical choice is the Gaussian kernel (in Eq. 3.3).

Batch algorithm

One possibility to implement a *batch SOM algorithm* is to add an extra step to the batch K-means procedure (Kohonen et al., 2000). After step 2 of the K-means procedure in Sec. 3.2.9, a smoothing pass along the grid is carried out:

$$\mathbf{c}_i := \frac{\sum_{j=1}^{K} |C_j| h(d_{ij}, \sigma(t)) \mathbf{c}_j}{\sum_{j=1}^{K} |C_j| h(d_{ij}, \sigma(t))}, \forall i \tag{3.16}$$

A relatively large neighborhood radius in the beginning gives a global ordering for the map. The kernel width $\sigma(t)$ is then decreased monotonically along with iteration steps which increases the flexibility of the map to provide lower quantization error in the end. If the radius is run to zero, the batch SOM becomes identical to K-means. Details of initialization, suggestions of learning parameter and neighborhood kernel selection are omitted here, see e.g., (Kohonen, 2001).

The batch SOM is a computational short-cut version of the basic, incremental SOM that was introduced by Kohonen (1982). Despite the intuitive clarity and elegance of the basic SOM, its mathematical analysis has turned out to be rather complex, see, e.g., (Flanagan, 2001). This comes from the fact that there exists no cost function that the basic SOM would minimize for a probability distribution (Erwin et al., 1992).

Some authors regard generative mixture models obtained by the expectation-minimization algorithm as a probabilistic counterpart of the K-means algorithm (Bottou and Bengio, 1995). In similar fashion, self-organizing structures having a probabilistic framework have been proposed (Bishop et al., 1998). Lampinen and Kostiainen (2002) present an add-on probability density model for the basic SOM.

In general, the number of map codebook vectors governs the computational complexity of one iteration step of the SOM. If the size of the SOM is scaled linearly with the number of data vectors, the load scales to $O(MN^2)$ (Kohonen et al., 2000). But on the other hand, the selection of K can be made following, e.g., $\sqrt{N}$ as suggested in (Vesanto, 2000), and the load decreases to $O(MN^{1.5})$. Vesanto (2000) considers details of the computational load of the implementation of the *SOM Toolbox*. It is suggested that the *SOM Toolbox* applies to small to medium data sets up to, say, 10 000–100 000 records. A specific problem is that the memory consumption in the *SOM Toolbox* grows quadratically along with the map size K.

Applications

In practice, the SOM and its variants (Kangas et al., 1990) have been successful in a considerable number of application fields and individual applications (Kaski et al., 1998a; Oja et al., 2003). In the context of this thesis, interesting application areas close to VDM include

- visualization and UI techniques especially in information retrieval, and exploratory data analysis in general (Ainsworth, 1998; Aristide, 1993; Honkela et al., 1996; Kaski, 1997; Kaski et al., 1998b, 1999, 2000, 2003b; König and Michel, 2003; Kohonen et al., 2000; Laaksonen et al., 2000; Nikkilä et al., 2002; Pampalk et al., 2002a,b, 2003; Rubio and Giménez, 2003; Ultsch, 2003; Vesanto and Ahola, 1999; Vesanto, 1999; Vesanto and Hollmén, 2002; Yang et al., 2003),
- context-aware computing (Schmidt et al., 1999a; Laerhoven et al., 2001)
- industrial applications for process monitoring and analysis (Alhoniemi et al., 1999; Laine, 2003; Simula et al., 1996).

Visualization capabilities, data and noise reduction by topologically restricted vector quantization, and practical robustness of the SOM are of benefit to data mining (Kaski, 1997; Vesanto, 2002). There are also methods for additional speed-ups in the SOM for especially large datasets in data mining (Lawrence et al., 1999) and in document retrieval applications (Kohonen et al., 2000).

The SOM framework is not restricted to Euclidean space or real vectors. For example, within the scope of visualization, Ritter (1999) presents a variant of the SOM in a non-Euclidean space to enhance modeling and visualizations of hierarchically distributed data. This method uses a fisheye distortion in the visualization. Also self-organizing maps and similar structures for symbolic data exist (Kohonen and Somervuo, 1998; Flanagan, 2003), and have been applied also to context-aware computation (Sec. 4.3) (Flanagan et al., 2002).

3.2.13 Some notes on SOM visualization

Compact, ordered, and trustworthy view to data

In visual data mining, the grid structure of the SOM gives a platform for visualization UI tool that is based on the topological order of the codebook vectors. The grid can then be used to present some other visualization primitives, different labels, glyphs, other data graphics, and UI tools, and others in an orderly fashion. Kaski (1997), Vesanto (1999), and Vesanto (2002) give good general overview of several aspects and methods related to SOM visualization.

Different visualizations of a SOM can be juxtaposed, see Fig. 3.5. Juxtaposing ordered views helps in comparing entities (cf. technique of small multiples (Tufte, 1983)). This is a common graphical technique used, e.g., in scatterplot matrices. Furthermore, compared to the proximity preserving projections with space between the markers, the SOM makes effective use of the visualization area since it automatically scales with the density of the data. Finally, Venna and Kaski (2001), Nikkilä et al. (2002), and Kaski et al. (2003a) have compared several proximity preserving projections and concluded that the SOM compares very favorably to other methods in retaining trustworthiness.

Cluster visualization

While being beneficial to efficient use of visualization space, the uniform grid representation of the SOM does not show any clustering in itself but visual cues for clustering have to be added to it otherwise.

A popular way to visualize clustering is to compute the distance between units in the data space and present the result as a *U-matrix* (Iivarinen et al., 1994; Ultsch and Siemon, 1990). To be successful, this visual means to detect clusters in the SOM require substantially more map units than anticipated clusters. A gray level coding of the U-matrix is presented in Fig. 3.5. Also, the average distance of the map unit and its neighbors can be shown on the SOM grid (Pampalk et al., 2002b; Vesanto, 1999). Contour lines can be used instead of gray level coding (Pampalk et al., 2002b). The SOM in document visualization has been approached from a cartographer's viewpoint as well (Skupin, 2002). Indeed, the U-matrix is strongly associated with a geographical map, e.g., "islands", mountain and valleys can be thought of as metaphors. Enhancements to the U-matrix and different techniques aiming at the same visual metaphor are presented, e.g, in (Kaski et al., 2000; Pampalk et al., 2002b; Ultsch, 2003).

Another possibility is to actually map the SOM unit distances in data space to distances between the visual markers representing the map units. In general, this cannot be done perfectly. One solution is to use a non-linear proximity preserving projection such as Sammon's projection, but on the other hand, the orderliness of the map may be lost. Kaski et al. (1999) describe a method that approximates the local distances but still maintains the orderliness of the SOM grid. The method directly optimizes an appropriate cost function by a stochastic gradient descent method.

The contraction projection explained in Publication 3 can also be used for the purpose of reflecting distances. It also incorporates an adjustable means of highlighting cluster structure in the data. When only a slight contraction is made, the map order is still maintained. When the contraction proceeds further, cluster structure is more highlighted. In contrast to the method of (Kaski et al., 1999), the contraction projection does not rigorously minimize a cost function. It does not return a single projection but a sequence of them. This requires that the user inspects different stage(s) of the contraction process and decide which of them are of interest. Furthermore, it involves a user specified kernel width parameter. On one hand, this can be seen as a problem since the method is quite heuristic. On the other hand, the contraction projection can be used with an interactive UI where one can change the kernel width parameter and the stage of contraction to move from a visualization between a uniform SOM grid to a clustered view. Figure 3.6 shows some stages of a contraction projection for the SOM on the Iris data. Recently, Rubio and Giménez (2003) have proposed several enhancements to SOM visualization, e.g., a "grouping neuron" method intended for a similar purpose as the contraction projection. Rubio and Giménez (2003) compare the grouping neuron method to the contraction projection presented in Publication 3 and suggest that their method is computationally more effective on larger SOMs. Furthermore, enhancing the global distance preservation by directly modifying the learning algorithm of SOM have been presented in (König and Michel, 2003; Yin, 2001).

The dissimilarities between the units can also be visualized as a color difference (Kaski et al., 1998b, 1999). Publication 2 demonstrates this using Sammon's projection of the map, and Publication 3 utilizes the SOM contraction projection to create a tunable color coding. The color coding in Publications 2 and 3 is based on mapping attribute values directly into the monitor RGB color space (Ware and Beatty, 1988; Ainsworth, 1998). Kaski et al. (1999) use visually more uniform CIELab color space, see, e.g., (Ware, 2000). When using CIELab color space, the mapping needs more consideration in order to cope with the shape of the color space and the output device gamut. Using direct RGB coding is practical in the sense that it skips these problems, and it has been found to work reasonably in cluster visualization—yet the visual non-uniformity degrades results in certain directions (Ware and Beatty, 1988).

Component planes and scatterplots

The problem with scatterplot matrices is that the number of pairwise comparisons grow quadratically. To some extent, the SOM *component planes* (Tryba et al., 1989) facilitate this problem. The component planes code the attribute values in different parts of the SOM using some retinal attribute, typically gray level, see Fig. 3.5. Thus, the component planes rely on comparing differences in the shape of the contours, and the notes on using gray level or color code in representing data values on surfaces apply here.

The visual search for similarities and differences among component planes can still be cumbersome. To ease the comparison task, Vesanto and Ahola (1999) have proposed a tool for ordering the component plane according to the global linear correlations between the variables. This helps comparing the component planes since roughly similar ones are grouped together. Detecting linear dependencies is, of course, not new. The added value of the SOM is in detecting partial correlations or clusters that appear, e.g., as spots in a subset of the component planes, see Publication 1. A problem is that the overfitting and folding of the map may introduce spurious correlations even in random data (Lampinen and Kostiainen, 2002).

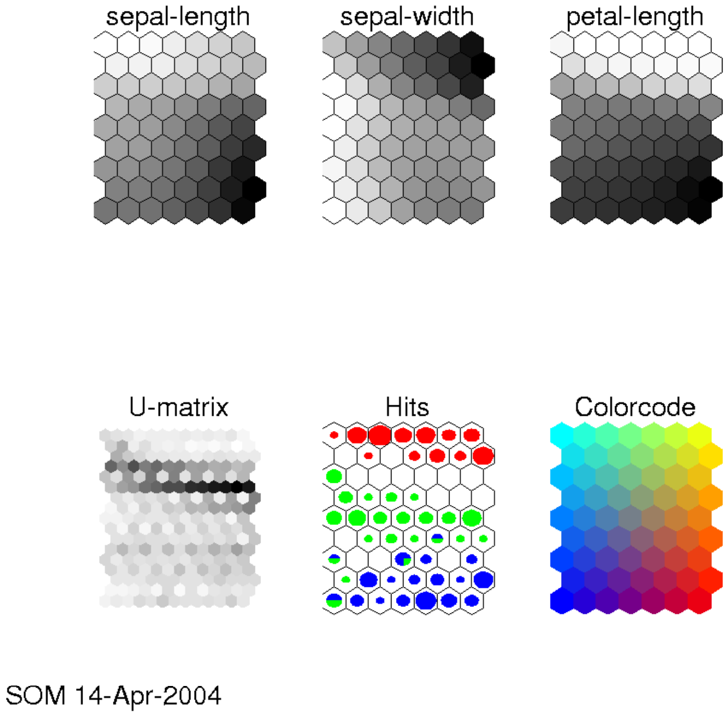
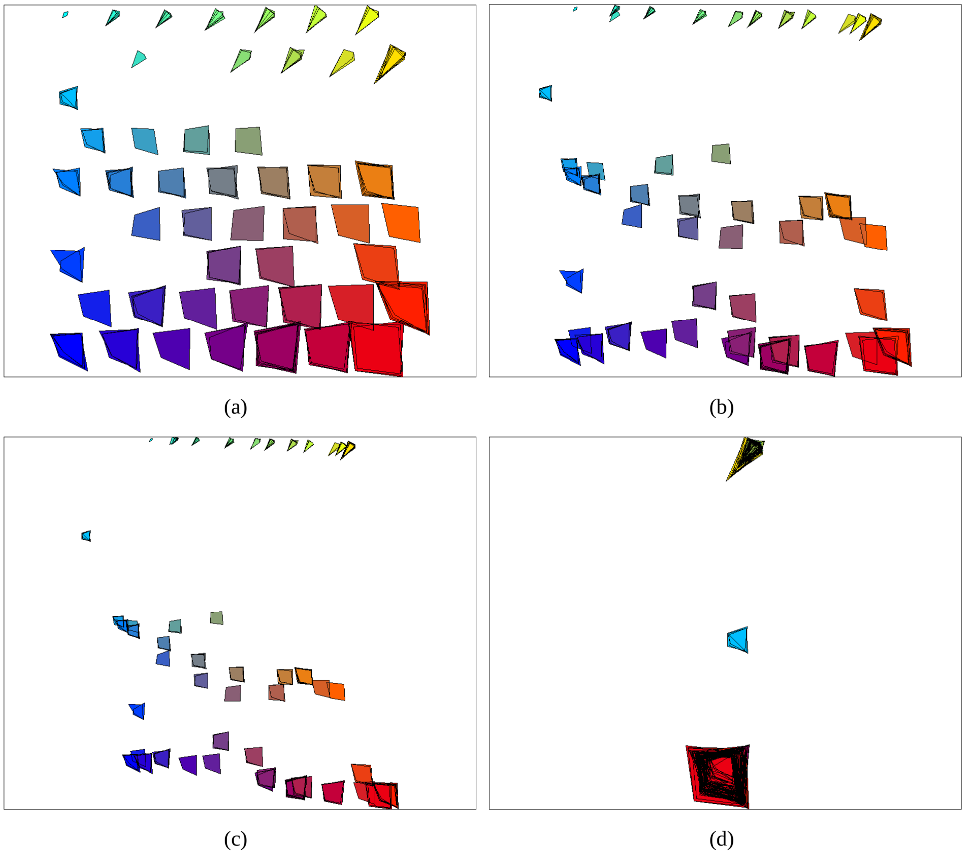


Figure 3.5: A typical presentation of a SOM in the *SOM Toolbox*. The SOM is made on the Iris data. The first row of panels shows the component planes. The gray level of each unit (hexagon) shows the value of the corresponding map codebook vector element. The first panel on the second row is the U-matrix of the map: the darker the gray, the longer the inter-unit distance. The second panel on the second row show, using pie diagrams, the amount (size) and share (color) of each class of data vectors projected in the map unit. The third panel on the second row is the color coding of the map units used in Figs. 2.4, 3.1, and 3.6. where the markers for data items get the color of *the best matching unit* in this SOM. The best matching unit for a data item is the one into which the data item gets projected, i.e., the closest unit in the data space.

Linking and brushing

The component planes and U-matrix presentation of SOM can be used in conjunction with other presentations of data by utilizing two common techniques in visualization: linking and brushing. For



(a)

(b)

(c)

(d)

Figure 3.6: Some different phases of a contraction projection on the SOM in Fig. 3.5. Panel (a) shows glyphs of the data on the best matching units on the original uniform SOM grid. The rest of the panels show phases of the contraction process ending into two major clusters.

linking, a SOM is given some ordered coding using some retinal attribute(s), e.g., color. The color of the best matching unit (see Fig. 3.5) for a data item is then transferred onto some other presentation of the same data. Aristide (1993) has demonstrated linking the SOM with PCA scatterplots, hierarchical clustering methods, and coloring the map of Europe according to the SOM presenting economical status of the countries. Kaski et al. (1998b) map economical status of countries using the SOM onto a world map using color coding. Ainsworth (1998) colors an oceanographic map using a 3D SOM trained with remote sensing data.[10]

Publications 1 and 2 apply the color linking between U-matrix, component planes, and ordinary scatterplots to process data analysis. Alhoniemi (2000) utilizes this method in investigation of pulping data. Furthermore, in Publication 1, an interactive brushing tool for SOM is shortly described. It was developed for exploring, especially, industrial process data.

[10] In this case, the colors come from the 3D SOM grid coordinates mapped directly into an RGB cube.

3.2.14 Clustering validity indices

The clustering methods in this thesis do not directly make a decision of the number of clusters but require it as a parameter. This poses a question which number of clusters fits best to the "natural structure" of the data. The problem is somewhat vaguely defined since the utility of clusters is not explicitly stated with any cost function. An approach to solve this, adopted in Publication 4, are the "add-on" relative clustering validity criteria. Basically, one clusters first the data with an algorithm with cluster number $K = 2, 3, \ldots, K_{max}$. Then, the index is computed for the partitions, and (local) minima, maxima, or knee of the index plot indicate the adequate choice(s) of K. The good news is that many of the relative cluster validity indices are fast to compute. The bad news, not surprisingly, is that the results of empirical studies with known number of clusters do not unanimously support a certain index for all data types (Bandyopadhyay and Maulik, 2001; Bezdek and Pal, 1998; Halkidi et al., 2001; Maulik and Bandyopadhay, 2002), or conclude after empirical tests that some classic indices hardly make any sense on the application data at hand (Vuori and Laaksonen, 2002).

Two examples of such indices appear in Publication 4 Davies-Bouldin type indices (Davies and Bouldin, 1979) are among the most popular relative clustering validity criteria:

$$I_{DB} = \frac{1}{K}\sum_{i=1}^{K} R_i, \quad R_i = \max \frac{\Delta(C_i) + \Delta(C_j)}{\delta(C_i, C_j)}, \quad \forall j, j \neq i \tag{3.17}$$

where $\Delta(C_i)$ is some adequate scalar measure for within-cluster dispersion and $\delta(C_i, C_j)$ for between-cluster dispersion. A simplified variant of this, the R-index (I_R), adopted from Levine and Domany (2001), was considered in Publication 4.

$$I_R = \frac{1}{K}\sum_{k=1}^{K} \frac{S_k^{in}}{S_k^{ex}}, \text{ where} \tag{3.18}$$

$$S_k^{in} = \frac{1}{|C_k|^2} \sum_{i,j \in C_k} d_{ij}, \quad \text{and} \quad S_k^{ex} = \min_l \frac{1}{|C_k||C_l|} \sum_{i \in C_k} \sum_{j \in C_l} d_{ij}, \quad (l \neq k).$$

In preliminary experiments, the R-index gave reasonable suggestions for a sensible number of clusters with a given benchmarking data set. It was compared to the Dunn-like indices ν_{AB} (Bezdek and Pal, 1998):

$$\nu_{AB} = \underbrace{\min}_{k, k \neq l} \left\{ \frac{\delta_A(C_k, C_l)}{\underbrace{\max}_{m} \Delta_B(C_m)} \right\} \tag{3.19}$$

where δ_A is some between-cluster dissimilarity measure; those defined in Eqs. 3.13–3.15 apply here, and Δ_B is some measure of within-cluster dispersion (diameter), e.g.,

$$\Delta_1(C_k) = \max d_{ij}, \quad i, j \in C_k \tag{3.20}$$

$$\Delta_2(C_k) = \frac{1}{|C_k|^2 - |C_k|} \sum_{i,j \in C_k} d_{ij}. \tag{3.21}$$

With this notation ν_{11} in Eq. 3.19 would be the original Dunn's index (Dunn, 1974). Davies-Bouldin and Dunn-like indices share a problem: if there is only one item in a cluster, or if the items are identical, the within-cluster scatter vanishes and this makes the index dubious when one-item clusters appear. There are literally dozens of relative cluster validity indices (Halkidi et al., 2001), and as is obvious, the selection of the R-index in Publication 4 is hardly optimal but a working solution and it is only meant to roughly guide the exploration.

Within the scope of time series segmentation, e.g, Vasko and Toivonen (2002) have studied selecting the proper number of segments using hypothesis testing based on random permutations and Levine and Domany (2001) have used random sampling for validating hierarchical clustering.

The clustering validity index, in the sense of model selection, is accompanied with tasks of

– checking the general clustering tendency, i.e., testing whether the hypothesis of random structure in data should abandoned in the first place

– external validity tests, i.e., comparing the clustering with some external partition, e.g., known classification.

For example, Jain and Dubes (1988) and Halkidi et al. (2001) discuss further different clustering evaluation tasks and methods.

Selecting the "best" SOM is more complicated as it differs very much from the basic clustering. Measuring the quality of a SOM in the sense of topological ordering is considered in (Kaski and Lagus, 1996; Kiviluoto, 1995; Polani, 2002).

3.2.15 Some further aspects of clustering

Large datasets and dimensionality

In this thesis, we restrict ourselves to quite small data sets with a modest number of dimensions (say, less than 50) However, often in data mining, large databases and very high-dimensional data are encountered. Consequently, computational problems and problems of dimensionality grow. Large databases require sophisticated clustering methods. For example, DENCLUE (Hinneburg and Keim, 1998) and BIRCH (Zhang et al., 1996) combine different clustering approaches along with scaling up techniques and techniques to accommodate noise. Halkidi et al. (2001) list further such systems and their characteristics. In addition to large number of samples, a big dimension can cause computational problems, one solution being dimension reduction by random projections utilized in (Bingham, 2003; Kohonen et al., 2000).

Feature selection and weighting

When data dimensionality is high, methods that rely on computing proximities using all attributes may fail since the contribution of task-irrelevant variables cause a noise-like effect that degrades the result. Traditionally, forward-backward attribute selection and global attribute weighting methods have been applied in selecting a subset of attributes that produces a "good" clustering (Fowlkes et al., 1988; Milligan, 1989). Furthermore, in some data there might be clusters in different subspaces, or to put it otherwise, different features (variables) may be unequally important for formation of different clusters. Detecting clusters in (low-dimensional) subspaces of high-dimensional data is the aim in *projected clustering* or *subspace clustering* (Aggarwal et al., 1999).

To sum up, the problem to select a proper proximity measure or set of features makes one believe that clustering is a tricky problem. Explicit introduction of some external criterion and information would surely benefit the exploration. For example, *discriminative clustering* (Sinkkonen and Kaski, 2002; Kaski et al., 2003b) is a clustering paradigm that utilizes prior class labels of the data items as an auxiliary constraint. As a result, the clustering reflects the features that are most relevant with respect to the prior classes of data (Sinkkonen, 2003). A different approach is a VDM style cluster analysis system for aiding an expert to make decisions: Laine (2003) and Pampalk et al. (2003) present SOM based VDM tools that facilitate feature selection and allow the effects of changing features in cluster structures to be seen. This kind of approach comes close to the line of exploratory data analysis described in Publication 1.

3.3 Finding interesting linear projections

Finding patterns in data can be assisted by searching an informative recoding of the original variables by a linear transformation. The linearity is at the same time the power and the weakness of these methods. On one hand, a linear model is limited, but on the other hand, potentially both computationally more tractable and intuitively more understandable than a non-linear method.

The use of PCA in this manner was explained in Sec. 3.2.4. However, searching for directions of maximum variance is not necessarily an interesting way of recoding the data. In exploratory *projection pursuit*, the rather broad intention is to maximize an index of *interestingness* of the variables (Friedman and Tukey, 1974). One measure of being interesting is that the distribution of a variable is far from being Gaussian (Hyvärinen et al., 2001). *Independent component analysis* (ICA) (Jutten and Herault, 1991; Hyvärinen et al., 2001) is based on a statistical model where the observed data is expressed as a linear transformation of latent variables that are non-Gaussian and mutually independent. In exploratory data analysis, ICA can be used to find interesting recoding of the data variables into attributes that might relate to or reveal some phenomena better than the original ones.

3.3.1 Independent component analysis

In the basic, linear and noise-free, ICA model, we have M latent variables s_i, i.e., the unknown *independent components* (or source signals) that are mixed linearly to form M observed signals, variables x_i. When $\mathbf{X}$ is the observed data, the model becomes

$$\mathbf{X} = \mathbf{AS} \tag{3.22}$$

where $\mathbf{A}$ is an unknown constant matrix, called the *mixing matrix*, and $\mathbf{S}$ contains the unknown independent components[11]; $\mathbf{S} = [\mathbf{s}(1) \quad \mathbf{s}(2) \ldots \mathbf{s}(N)]$ consisting of vectors $\mathbf{s}(i)$, $\mathbf{s} = [s_1 \quad s_2 \ldots s_M]^T$. The task is to estimate the mixing matrix $\mathbf{A}$ (and the realizations of the independent components s_i) using the observed data $\mathbf{X}$ alone. Comon (1994) presents exact conditions for the identifiability of the model; the most fundamental is that the independent components must have non-Gaussian distributions. However, what is often estimated in practice, is the *demixing matrix* $\mathbf{W}$ for $\mathbf{S} = \mathbf{WX}$, where $\mathbf{W}$ is a (pseudo)inverse of $\mathbf{A}$ (Hyvärinen et al., 2001).

This kind of problem setting is pronounced in *blind signal separation* (BSS) problems, such as the “cocktail party problem” where one has to resolve the utterance of many nearby speakers in the same room. Publication 4 gives a typical example of BSS where the benchmarking task for ICA is the separation of brain and noise signal sources.

Several algorithms for performing ICA have been proposed, derived from different theoretical grounds (Hyvärinen et al., 2001). In Publications 4, 5, and 6, the FastICA algorithm is used. This fast, fixed point algorithm is briefly described in the next section.

3.3.2 FastICA and *Icasso*

The FastICA algorithm (Hyvärinen and Oja, 1997) is based on finding projections that maximize non-Gaussianity measured by an objective function J. *Kurtosis* $E(y^4) - 3\left(E(y^2)\right)^2$(applies to a zero mean variable y) was originally used as a basis of the objective function. It is somewhat sensitive to outliers, and (Hyvärinen, 1999) proposes alternative, more robust objective functions.

A necessary condition for independence is uncorrelatedness, and a way of making the basic ICA problem somewhat easier is to whiten the original signals $\mathbf{X}$. Thereafter, it suffices to rotate the whitened data $\mathbf{Z}$ suitably, i.e., to find an orthogonal demixing matrix that produces the estimates for

[11] A row of $\mathbf{S}$ is an independent component.

the independent components $\mathbf{S} = \mathbf{W}^*\mathbf{Z}$ (Hyvärinen et al., 2001). When the whitening is performed by Eq. 3.6 the demixing matrix for the original, centered data is $\mathbf{W} = \mathbf{W}^*\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{E}^T$.

Here, we present the symmetrical version of the FastICA algorithm where all independent components are estimated simultaneously:

1. Whiten the data. For simplicity, we denote here the whitened data vectors by $\mathbf{x}$ and the mixing matrix for whitened data with $\mathbf{W}$.
2. Initialize the demixing matrix $\mathbf{W} = [\mathbf{w}_1^T \quad \mathbf{w}_2^T \dots \mathbf{w}_M^T]$, e.g., randomly.
3. Compute new basis vectors using update rule
   $\mathbf{w}_j := E\left(g(\mathbf{w}_j^T\mathbf{x})\mathbf{x}\right) - E\left(g'(\mathbf{w}_j^T\mathbf{x})\right)\mathbf{w}_j$
   where g is a non-linearity derived from the objective function J; in case of kurtosis it becomes $g(u) = u^3$, and in case of skewness $g(u) = u^2$ (Hyvärinen et al., 2001). Use sample estimates for expectations.
4. Orthogonalize the new $\mathbf{W}$, e.g., by $\mathbf{W} := \mathbf{W}(\mathbf{W}^T\mathbf{W})^{-1/2}$.
5. Repeat from step 3 until convergence.

There is also a deflatory version of the FastICA algorithm that finds the independent components one by one. It searches for a new component by using the fixed point iteration (in step 3 of the procedure above) in the remaining subspace that is orthogonal to previously found estimates.

Both practical and theoretical reasons make the FastICA an appealing algorithm. It has very competitive computational and convergence properties. Furthermore, FastICA is not restricted to resolve either super- or sub-Gaussian sources of the original sources as it is the case with many algorithms (Hyvärinen et al., 2001).[12] However, the FastICA algorithm faces the same problems related to suboptimal local minima and random initialization which appear in many other algorithms—including K-means and GIR. Consequently, a special tool *Icasso* for VDM style assessment of the results was developed in the course of this thesis in Publication 4.

A recent method by Meinecke et al. (2002) is intended for the same purpose as *Icasso* and it is also based on resampling. The method of Meinecke et al. (2002) first estimates the independent components for the original data. To our understanding, these estimates act then as fixed cluster centers during the rest of the process. After computing the initial estimates, bootstrapping (Efron and Tibshirami, 1998), is used to introduce perturbation to the resampled estimates. The bootstrapped ICA estimates are clustered, using the initial estimates as cluster centroids. Dispersion in the clustered estimates and the confusion between them is finally computed.

The procedure of Meinecke et al. (2002) works fine for an algorithm that has no stochastic elements—being stable in the sense that it always gives the same result for the same data. An example of such an ICA algorithm is JADE[13] (Cardoso and Soloumiac, 1993) used in (Meinecke et al., 2002). However, this is surely not the case with FastICA with a random initial condition. We assume that if proceeding as Meinecke et al. (2002), a single, possibly "bad" set of initial estimates from FastICA could form unnatural clusters when used as fixed centers.[14] Therefore, *Icasso* pools all the estimates together and forms the clusters bottom-up among them. In addition, *Icasso* is concentrated in UI and visualization of the estimate space by means of a non-linear mapping. The approach can

[12] A super-Gaussian variable has positive kurtosis: distribution has a sharp peak and heavy tails. A sub-Gaussian variable has negative kurtosis: distribution is flat with light tails.

[13] Joint Approximation of Diagonal Eigenmatrices

[14] We recently found out that there is an implementation of this method at URL `http://ida.first.fraunhofer.de/~meinecke/download_ica_resamp.html` [referred Apr, 2004] which could be modified to use FastICA to compare the methods and test our assumption.

be used in other similar applications—only a proximity matrix between the estimate is needed. One could, e.g., investigate the results of K-means in this fashion.

3.3.3 ICA and binary mixture of binary signals

Next, we consider a very specific non-linear mixture of latent variables, the problem of the Boolean mixture of latent binary signals and possibly binary noise. Typically, prior work with ICA and binary data concentrates on linear mixtures of discrete (binary) sources and Gaussian noise, e.g., (Pajunen, 1997; Palmieri et al., 1998). See (Bingham, 2003) for further discussion and references. In general, non-linear extensions of ICA is a broad area of work, and "non-linear ICA" is highly non-unique as a problem (Hyvärinen and Pajunen, 1999).

The mixing matrix $\mathbf{A}^B$, the observed data vectors $\mathbf{x}^B$ and the independent, latent source vectors $\mathbf{s}^B$ all consist now of binary vectors $\in \{0,1\}^M$. The basic model in Eq. 3.22 is replaced by a Boolean expression

$$x_i^B = \bigvee_{j=1}^{n} a_{ij}^B \wedge s_j^B, \quad i = 1,2,\ldots,M \tag{3.23}$$

where $\wedge$ is Boolean AND and $\vee$ Boolean OR. Instead of using Boolean operators Eq. 3.23 could be written $\mathbf{x}^B = U(\mathbf{A}^B\mathbf{s}^B)$ using a step function U as a post-mixture non-linearity (Taleb and Jutten, 1999). The mixture can be further corrupted by binary noise: exclusive-OR type of noise is considered in Publication 6.

On one hand, the basic ICA cannot solve the problem in Eq. 3.23 in general (Bingham, 2003). The methods for post-non-linear mixtures (Taleb and Jutten, 1999) that assume invertible non-linearity cannot be directly applied either. On the other hand, it seems possible that the basic ICA could work for data emerging from sources and basis vectors that are "sparse enough". Consequently, we experimented in Publication 6 how far the performance of the basic ICA can be pushed, using reasonable heuristics, without elaborating something completely new. In this thesis, the experiment can be seen as a feasibility study for using ICA as done in Publication 5 where the data was close to binary. In the context-awareness applications of this thesis, purely binary sources, observed values, and mixing process could surely be expected. Furthermore, there are similar problems in other application fields, prominently in text document analysis where such data is encountered. Bingham (2003) surveys approaches related to the binary latent variable problem, especially within the scope of document analysis. Since the basic ICA model is not the optimal choice for handling such problems in general, probabilistic models and algorithms have recently been developed for this purpose (Bingham et al., 2002; Seppänen et al., 2003).

In the procedure of Publication 6, the estimated linear mixing matrix $\hat{\mathbf{A}}$ is normalized by dividing each column with the element whose magnitude is largest in that column. Second, the elements below and equal to 0.5 are rounded to zero and those above 0.5 to one:

$$\hat{\mathbf{A}}^B = U(\hat{\mathbf{A}}\mathbf{\Lambda} - \mathbf{T}) \tag{3.24}$$

where the diagonal scaling matrix $\mathbf{\Lambda}$ has elements

$$\lambda_i = \frac{1}{\mathrm{smax}(\hat{\mathbf{a}}_i)} \text{ where} \tag{3.25}$$

$$\mathrm{smax}(\hat{\mathbf{a}}_i) = \begin{cases} \min \hat{\mathbf{a}}_i & \text{if } |\min \hat{\mathbf{a}}_i| > |\max \hat{\mathbf{a}}_i| \\ \max \hat{\mathbf{a}}_i & \text{otherwise.} \end{cases}$$

where $\max \hat{\mathbf{a}}_i$ means taking the maximum and $\min \hat{\mathbf{a}}_i$ the minimum element of the column vector $\hat{\mathbf{a}}_i$. Matrix **T** contains thresholds, here we set $t_{ij} = 0.5, \forall i, j$.[15] As supposed, this trick works quite well with sparse data and *skewness* $E(y^3)$ works better than kurtosis as a basis for the objective function on a wide range of sparsity data, except for noisy data.

But still, why does the basic ICA not give substantial results in Publication 5? Two reasons can be considered: Firstly, it may well be that the context data described in Publications 5 and 7 does not emerge from a model that is close enough to what can be solved with basic (or thresholded) ICA. Secondly, although we see a clear temporal dependence between events in the signals, the result of ICA estimation is permutation invariant since the model is an instantaneous (memoryless) mixture. To sum up, an independent latent variable model is potentially useful for the context recognition problem, but the data at hand requires more effort. Supposedly, an instantaneous mixing model of the data is not enough but the time structure has to be considered as well.

[15] Note that there are typos in Eq. 4 of Publication 6 and it does not correspond to the scaling proper made in the experiments. Here, Eqs. 3.24 and 3.25 give the correct procedure that was used.

Chapter 4

Data mining and user interfaces

4.1 Emerging user interface paradigms

Context-awareness

Publications 5 and 7 are related to *context-aware computing* (Schilit et al., 1994) which has become recently a major topic in human-computer interaction studies (Moran and Dourish, 2001). Context-aware computing involves adaptive/intelligent UIs and other applications that react to or benefit from the context intelligently. It is related to the framework of ubiquitous (pervasive) computing (Weiser, 1991) and often appears together with “wearable computers” (Farringdon et al., 1999; Golding and Lesh, 1999; Laerhoven and Cakmakci, 2000; Laerhoven et al., 2002; Pascoe, 1998) and other emerging research fields of human-computer interaction (Moran and Dourish, 2001). A more conservative, user-centric approach to modeling context is taken in usability studies where one analyzes and models the context of use in advance, e.g., by field or laboratory tests, and uses that knowledge in interface and product design (Holtzblatt, 2003; Maguire, 2001).

Sensor-based mobile-awareness

We concentrate more specifically on *mobile-aware applications* (Chen and Kotz, 2000). The benefits of context-awareness are prominent in mobile communications where the communication situations can vary a lot. The mobile terminals are becoming more and more sophisticated in their function yet smaller in their size. User interaction could be made easier and less intruding if the mobile terminal recognized the user’s current context and adapted its functions accordingly without constant attention from the user (Chen and Kotz, 2000). In particular, Publications 5 and 7 focus on *sensor-based mobile-awareness* (Clarkson et al., 2000; Golding and Lesh, 1999; Mäntyjärvi, 2003; Schmidt et al., 1999a).

Personalization

Many issues in mobile-aware UIs have to do with *personalization* as well. Blom and Monk (2003) have studied and discussed user-centric personalization of appearance, and use the following definition of personalization (of technology): “…a process that changes the functionality, interface, information content or distinctiveness of a system to increase its personal relevance to an individual”.

The adaptive keyboard in Publication 8 is a straight-forward application in the field of personalization and adaptive UIs. In general, the effects of personalization are not confined to cognitive ease of use but include also aspects that are related to user's personality, emotions and group identity. For example, feeling of personality and ownership, reflection of personal or group identity, improved aesthetics, being-in-control, attachment to the system, or simply having change or fun are aspects of why people personalize their mobile phones (Blom and Monk, 2003).

Adaptive and multimodal interfaces

Mobility leads to rapid change of usage contexts and people tend to prefer different ways of using their personal mobile devices. This give rise to a growing need for user- and context-adaptivity—context-awareness and personalization relate to the wide field of *adaptive interfaces*. Jameson (2003) lists typical tasks of adaptive user interfaces: functions for *supporting the system use* (adapting the interface, automatizing routine tasks) or *information acquisition* (supporting browsing, spontaneous provision of information by situation). The same tasks appear in context-aware applications as well (cf. Sec. 4.2.2). Characteristically, context-awareness focuses in adapting to the context of use in addition to directly adapting to the user's preferences. We will not pursue usability in this thesis, but obviously, usability challenges of the adaptive UIs, such as predictability, transparency, controllability, unobtrusiveness, and privacy (Jameson, 2003), are essential in mobile-aware interaction as well.

Furthermore, using on-board sensors to determine user activity touches *multimodal interfaces* that process several combined user input modes in a coordinated manner (Oviatt, 2003). Though, this thesis does not consider explicit multimodal control of a device, we point out that the research challenges of context recognition (Sec. 4.3) are similar to those of multimodality: how to fuse information from several simultaneous and/or sequential input modes and events and how to integrate the inputs on feature level and at a higher, semantic level (Oviatt, 2003). In general, machine learning is needed to take use of non-explicit inputs from the different input modes and other user information (Jameson, 2003; Oviatt, 2003).

4.2 Context-awareness

4.2.1 What does "the context" mean—in this context?

In context-awareness studies, there are different operational definitions and categorizations of context. These definitions identify different context classes (or dimensions) and features of context-aware applications. A widely cited operational definition from Dey and Abowd (1999) and Dey et al. (2001) state that "context" is

> "... any information that can be used to characterize the situation of entities (i.e., whether a person, place, or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves. Context is typically the location, identity and state of people, groups, and computational and physical objects."

Golding and Lesh (1999) state shortly that context-awareness is "the ability to detect aspects of the user's internal or external state". Furthermore, Dey and Abowd (1999) define that context-awareness is about giving relevant information and/or services to the user where relevancy depends on the user's task and the context. These are obviously rather wide definitions; deducing what is "relevant information" is anything else but a simple task (Dourish, 2004; Greenberg, 2001; Mäntyjärvi et al., 2003).

We have based the formation of “contexts” on a bottom-up machine learning construction from low-level features. The approach is similar to that of Schmidt et al. (1999a) who consider “contexts” as higher-level abstraction, e.g., extracted patterns or decided rules, of the primitive features obtained from various information sources. We have adopted also the idea of using a variety of simple, cheap sensors for various phenomena, acceleration, light, sound, etc. (Golding and Lesh, 1999; Laerhoven et al., 2001; Schmidt et al., 1999b).

Schilit et al. (1994) recognize the following, three general types of contexts or environments:

1. *User context* describes the “human factors”, the situation from the user’s point of view, including the social situation, and the user’s status and activity. Location is an important and easily operationalized indicator of user context, but reflections of the user context can be found in phone profile settings, calendar markings, or current use of, e.g., communication devices. Setting explicit information sources, context tags, accessible in a short range network can be used to label important places and services (Want et al., 1992). Some aspects of the social context could also be inferred from an ad hoc network of mobile devices carried by the people nearby (Mäntyjärvi et al., 2002a).

2. *Physical context* of the user and the device is important in sensor-based mobile-awareness. It is assumed that the physical context can be used to gain information about the user context. In order to resolve the physical context, a piece of ambient information can be obtained by directly monitoring the user’s environment and actions, and possibly the user him/herself, using a set of on-board different sensors (Golding and Lesh, 1999; Schmidt et al., 1999b; Mäntyjärvi et al., 2003). The physical location can be obtained from a navigation system, like the Global Positioning System, or from the mobile network cell.

3. *Computing context* describes the world of the device itself: available connectivities, computation and application resources, networks, access to other devices and services etc. This is the dimension that we wish to make easy-to-use, or even invisible, for the user.

Obviously, *time* is also an important descriptor of the context. It can be associated to user or physical context. Chen and Kotz (2000) add time as a separate, fourth main category to the previous ones. Eventually, enumeration of dimensions or features is a rather simplistic approach and miss many dynamical aspects of context (Greenberg, 2001; Dourish, 2004). When one comes to think of it, context-*awareness*—if taken to the utmost—is about making an artificial intelligence that reacts reasonably in changing social surroundings, so the device should have commonsense-based knowledge on the world and human intentions (Minsky, 2000). However, the straight-forward thinking of prior, fixed elementary features is instrumental for our exploratory and constructive machine learning approach, something to start with.

4.2.2 Context-aware applications

Dey and Abowd (1999) characterize context-aware functionalities, based on earlier categorizations by Schilit et al. (1994) and Pascoe (1998), as follows:

1. *Automatic execution of a service* is the category that Schilit et al. (1994) identifies as *context-triggered actions*. A sensor-based example: if we know from the context information that the user is running, the font used in the display can be larger. Similarly, audio volume can be adjusted to compensate for higher levels of noise (Schmidt et al., 1999b; Mäntyjärvi and Seppänen, 2003). Another example of such an application is a context reminder (Schmidt et al., 1999b) that is triggered by a certain context. For example, the reminder “remember to buy milk” could be switched on near a grocery. Another typical example of context triggered

service is an automated profile change (Schmidt et al., 1999b), e.g., the phone is set into “silent” when the opera night begins.

2. *Presentation of information and services to user* includes sharing user context with others. For example, one can show one’s own context to others in order to facilitate communication with mobile phones (Schmidt et al., 2000). *Proximate selections* exemplifies also this class of functionalities, e.g., the user can be shown an adaptive list of nearby auxiliary devices such as printers (Schilit et al., 1994).

3. *Tagging of context to information for retrieval* (indexing by context). The segmentation of the context time series in Publication 7 can be understood as tagging the context stream for retrieval of interesting context changes. For example, in Fig. 4.2 the context is used to create a narrative of what happened: the user is not required to watch the whole video.

Development and widespread use of context-aware applications require common software tools and conceptualization (Dey et al., 2001), standardization, and consideration of privacy issues. A broad selection of issues on context-awareness and surveys on individual applications can be found in (Moran and Dourish, 2001) and other articles of the same volume, as well as in (Chen and Kotz, 2000; Mäntyjärvi et al., 2003; Dey and Abowd, 1999; Schmidt et al., 1999b).

4.3 Context data mining

4.3.1 Context recognition

This thesis does not present very specific context applications. The issue is *how* the context could be extracted from data in general, i.e., *the context recognition*. Mäntyjärvi (2003) defines context recognition as “the process of extracting, fusing and converting relevant data from [context] sources to a representation to be utilized in the application.” Publications 5 and 7 highlight the problem of context recognition as an interesting application for data mining and machine learning.

One solution for context recognition is a designed, rule based function, see e.g., (Mäntyjärvi and Seppänen, 2003). The knowledge base for such functions can be implemented by the system designer but also the user can customize the context-aware functionality with appropriate tools (Mäntyjärvi et al., 2003). There is, however, a problem in the previous approach: when the available set of context features, contexts and functions grow, it is not feasible anymore to expect the user or a system designer to be able to plan everything in advance (Greenberg, 2001). Also learning to use such systems can be cumbersome: Mäntyjärvi et al. (2003) report that users sometimes found it difficult to customize context dependent applications for themselves on a quite similar setting that appears in Publications 5 and 7.

In a supervised machine learning approach for context triggered actions, the device autonomously learns what actions or applications the user prefers in the recognized contexts and suggests automating some of these. A requirement for this kind of application to be successful is that the contexts can be classified. A naïve Bayesian classifier is applied to audio context features in (Korpipää et al., 2003) and to a more general set of sensor sources Golding and Lesh (1999) for indoor navigation. Different work in using multiple acceleration signals in gesture or body movement identification are given in (Mäntyjärvi et al., 2001b; Mäntylä et al., 2000; Laerhoven et al., 2002). (Clarkson and Pentland, 1999) and Clarkson et al. (2000) experiment with hidden Markov models to determine and classify the occurrence of audio-visual contextual events from a wearable video camera. Laerhoven et al. (2001) present a combination of a SOM and hidden Markov models for on-line recognition of contexts on sensor data in wearable computers.

In our unsupervised pattern recognition approach, the aim is to form contexts as higher-level attributes (patterns) by fusing simple and understandable, binary or fuzzy attributes (features) describing device position, user movements and ambient light and sound level. We try the feasibility of forming new attributes by using independent components and time-series segmentation. Principal components are used for exploratory stage to visualize and fuse context sources. At this stage, we do not present supervised recognition results, but explore the data and the methods.

Collecting information from on-board sensors (Publications 5 and 7) was quite simple and, in our study, the sensors are the only data sources that provide information on ambient context. In general, it would be rather unwise to stubbornly infer user context from ambient physical measurements alone and discard information obtained from the logs of different applications, e.g., calling, messaging, using calendar, or profiling. Moreover, we focus on autonomous context recognition system in the study, although specialized "context tags" and other context infrastructure embedded in the environment would be a valuable source of information in practice. Arranging the collection of all this additional information would have required considerable more effort and added the complexity of the study. For example, at the stage of this study the applications running in the phone and the UI actions could not be logged with reasonable effort. However, we assume that the same data-analysis methods can be extended to data sources other than the specific sensors used in this study.

4.3.2 Acquiring context data and features

In general, the set of sensors were selected so that their price, size and availability would be suitable to a mobile device—the criteria frequently used in sensor-based wearable computing. Furthermore, using video images was not considered. Working assumptions were that recording image of the environment is often not possible in typical use of mobile phone and that image recognition probably needs more processing power than touch, acceleration, and sound pressure signals. Identification of user movements and gestures based on acceleration signals is usual in context-awareness as explained earlier. In this study, rather simple features "stable"/"unstable" and "walking"/"running" are used. They were extracted based on the power of different frequency bands of the acceleration signal; Mäntyjärvi (2003) describes the relevant sensors and feature extraction methods in more detail.

In Publications 5 and 7, the context features are called "context atoms" to underline that they are given *a priori* and the rest of the context inference emerges from these.[1] The features have been selected so that they reflect everyday concepts. This eases understanding the meaning of the extracted context and integrating fuzzy inference or other rule based parts to the same system (Mäntyjärvi et al., 2001b, 2003). All context features are in the range $[0, 1]$, including both binary attributes and attributes that are fuzzy relations. For example, we say that ambient noise is "low", "medium", or "high" instead of presenting a single attribute. This eases combining various kinds of information sources that might be accessible. For example, calendar markings or semantic descriptions of locations like "meeting" or "Helsinki" would be inherently symbolic. Therefore, Himberg et al. (2003) and Flanagan et al. (2002) use the *symbolic clustering map* (Flanagan, 2003), a method specialized for symbolic data, to cluster context data.

4.3.3 Context recognition is context fusion

In this thesis, we confine to an off-line system where the context atoms are collected first, and the recognition is made on the data set. In an on-line system the context atoms would come in as a stream from which the frequently occurring or otherwise interesting patterns were inferred continuously. The

[1] Schmidt et al. (1999a) call the features derived from sensor signals *context cues*. Dey et al. (2001) use the name *context widget* for the software components that return preprocessed low-level context information from the sensors.

patterns, clusters or segments, are labeled with a new set of symbols that forms a higher-level context representation. Flanagan et al. (2002) present *instantaneous* and *sequential fusion* of context data as follows:

Instantaneous fusion

In *instantaneous fusion* a higher-level context is generated by the fusion of lower-level context attributes at time instant t. For example, simultaneous occurrence of “running” and “railway station” might imply “being late”. In practice, the same “true” context does not always produce exactly the same context attributes either due to variations in the environment or simply by the imperfection of feature extraction. However, one can characterize the data by clustering to K clusters, e.g., (Mäntyjärvi et al., 2001a; Flanagan et al., 2002). The clusters can be labeled with symbols which form a set of new, higher-level context attributes. The clustering tendency of the data is quite evident also in the visualization of Publication 5 presented in Fig. 4.1. See also Fig. 4.2 where the clustering algorithm of (Flanagan et al., 2002) is used to find six clusters in the data, and different UI profiles are associated with these clusters (contexts) (Himberg et al., 2003). (Note that the test scenario, and thus, the data, is not the same in Figs. 4.1 and 4.2.)

The basic clustering allows only one prototypic “context” to be active simultaneously. Publication 5 demonstrates the possibilities of PCA and ICA in context fusion (Salam and Erten, 1999). It is still an instantaneous fusion but, especially in the tests with ICA, the aim is to find several contexts (latent sources) that can be active independently at the same time. This is an important difference to the other approached surveyed in this thesis.

Sequential fusion

Enumerating instantaneous states as context does not always follow our figure of situations in everyday life. The order of events tell important information and this is lost if the temporal structure of data is ignored as in instantaneous fusion. *Sequential fusion* of context information source(s) into a higher-level context is about finding event episodes (Hand et al., 2001). For example, “browsing bus time table”,“walking-inside” succeeded by “walking-daylight” might imply “heading for a bus stop”. Segmentation of context data helps in finding such events as it highlights changes in the context. See Fig. 4.2.

The two modes of information fusion can be combined in a hierarchical manner and/or mixed. On one hand, the recognized clusters/segments serve as higher-level contexts that show which combinations of the basic features form common patterns in the data. On the other hand, the can be used as new features for the next level of recognition. Here, the context recognition is hierarchical context fusion.

Unsupervised learning is a general tool to begin with. The obvious problem with the unsupervised learning is that it lacks a unique, data driven way of deciding whether the extracted higher-level context, if any, are relevant. However, in Publications 5 and 7, the contexts and the extracted patterns are quite well aligned with common sense when the events in the video recordings of the text scenarios. If optimized predictions of preferred application(s) or service(s) and different context mapping are desired, some form of supervised learning would obviously help.

4.4 Adaptive keyboard

4.4.1 Text input methods for handheld devices

Text input methods for handheld devices are intensively studied since the small size of a mobile device challenges traditional keyboard design. Many different approaches have been proposed: a de facto standard in mobile phones seems to be the multi-tap method, possibly accelerated to “one-tap” with linguistic disambiguation technique, such as the predictive text input system T9.[2] Folded or miniaturized Qwerty or half-Qwerty keyboards are often feasible solutions (MacKenzie and Soukoreff, 2002).

Soft keyboards

Touchscreen—which is used in Publication 8—is a common choice to produce a soft keyboard and other input techniques. A potential alternative for producing a soft keyboard is an optically projected keyboard (Tomasi et al., 2003).[3] A remote sensing and pattern recognition method must be used to recognize which of the virtual keys the user types (Mäntyjärvi et al., 2002b; Tomasi et al., 2003). The problem with a typical soft keyboard is that it misses the sensations produced by the mechanical one which makes, e.g., eyes-free entry difficult (MacKenzie and Soukoreff, 2002). Up to some extent, this can be compensated with sound effects (Tomasi et al., 2003). However, the soft keyboard is very reconfigurable and gives potential to various novel input applications.

Optimal layouts for fast typing

By using soft keyboard, it is easy to reconfigure the keyboard layout according to the usage situation or other purposes. For example, there are different methods for completely reconfiguring a keyboard for fast typing on a specified language for a general user (MacKenzie and Soukoreff, 2002; Zhai et al., 2002). A generally and theoretically optimal layout for writing speed can be computed based on the linguistic statistics and quantitative models of user performance on such tasks (Fitts, 1954; Ward et al., 2002; Zhai et al., 2002). The predicted expert writing speed of best optimized layouts is reported to increase to 41–43 wpm—well over the soft, standard Qwerty keyboard having 30 wpm (MacKenzie and Soukoreff, 2002).[4]

Other manual input modes

Touchscreen makes it possible to create other manual input modes. These include recognizing handwriting or specialized graffiti, or strokes (MacKenzie and Soukoreff, 2002; Ward et al., 2002). Learning, adaptation and prediction are potential boosters for such methods also (Vuori et al., 2000; Vuori and Laaksonen, 2002; Ward et al., 2002). The acceleration sensor based recognition techniques can also be applied to manual input devices: Partridge et al. (2002) and Wigdor and Balakrishnan (2003) report methods for text input by tilting the device.

[2] T9 is a registered trademark of Tegic Communications, Inc.

[3] However, it seems that the projection technique in (Tomasi et al., 2003) only allows a fixed layout, still.

[4] Note that the predicted expert input speed on the commonly used one-tap with disambiguation also exceeds 40 wpm (Silfverberg et al., 2000). Such estimates are often model based *predictions* instead of empirical tests. Evaluation of such methods is difficult in general (MacKenzie and Soukoreff, 2002).

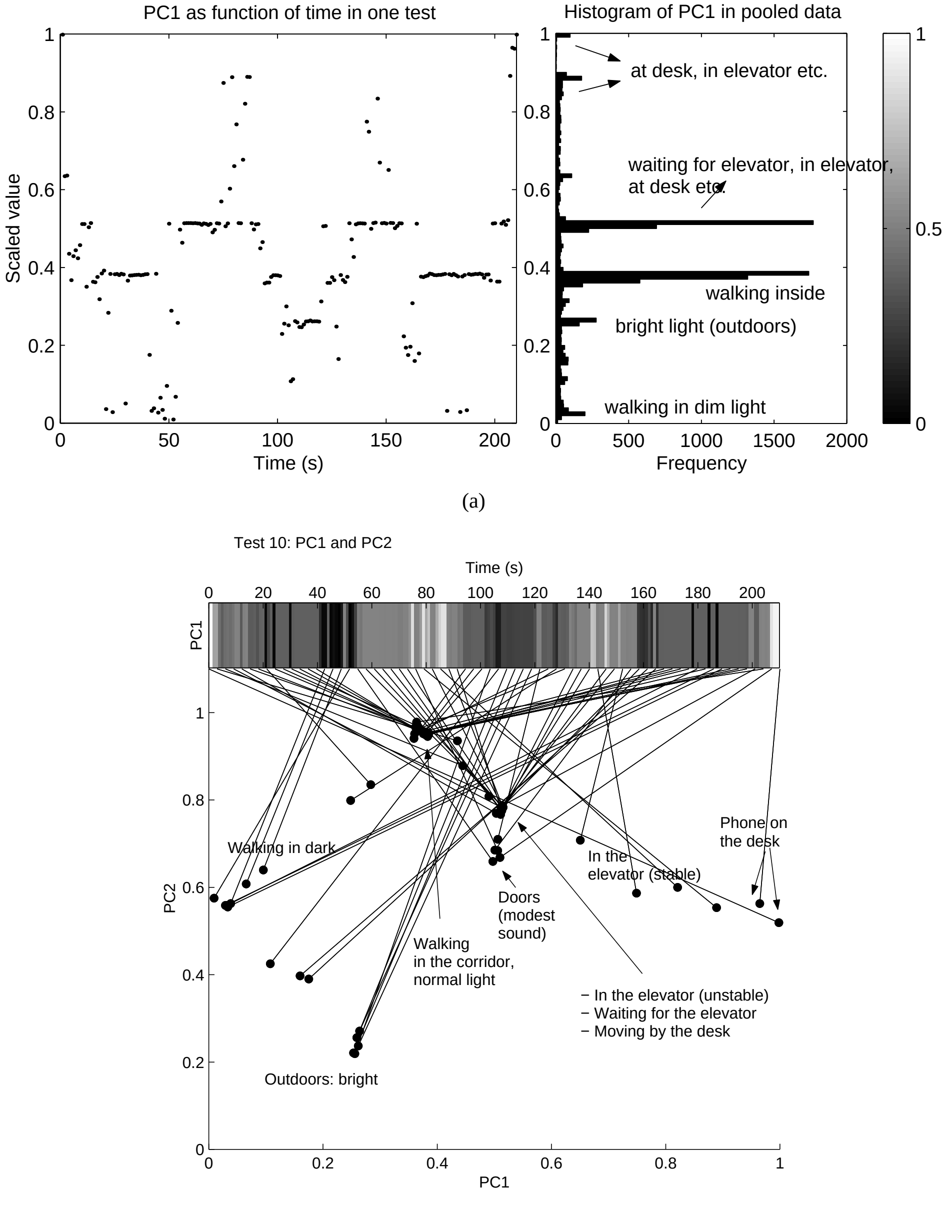
4.4.2 Adaptive personalization of a numeric keyboard

Publication 8 proposes a prototype of an “ergonomically” personalized keyboard design that would adapt to user’s personal style of typing but retain the general layout of the keyboard. Instead of proposing a completely new input method for a handheld device, we aim at slightly changing the conventional, numeric keyboard. The purpose is not to change the original global order of the keys but just to adjust the positions of the keys.

The individual physiology of the hand and personal fashion of typing are not captured by the general models typically used in keyboard optimization. In the absence of a well-established model, we started by exploring the problem based on a very simple learning technique.[5] The problem with this adaptation strategy is that the size of the keys may grow small unconstrained.

Based on the qualitative results, we believe that the idea of the personalized keyboard in Publication 8 is promising, but the same results indicate that the testing procedure should be extended—and the adaptation algorithm redesigned. For example, the effects of the learning curve and test procedure fatigue should be properly canceled. The users graded the adaptive keyboard positively, but such a grade might be upward biased. A proper evaluation of the potential success of such a feature would require also measuring other factors than typing speed. In fact, here the effects of novelty, potential upgrade of ergonomics, user feelings and the objective performance measures are intertwined.

[5] Anecdotally, this application goes in line with the clustering and visualization theme of this thesis. The idea of this kind of an adaptive keyboard came substantially from visualization of K-means algorithm on two dimensional data.



(a)

(b)

Figure 4.1: PCA visualizations on context data. The time series plot on panel (a) gives the values of the 1st principal component of part of the data. The bar chart in panel (a) gives the frequencies of the values of the 1st principal component in all data. A clear clustering structure is reflected already in this plot. One can say that certain value levels of the 1st principal component can be interpreted as contexts. Using PCA as a 2D proximity preserving projection is demonstrated in panel (b) using the 1st and 2nd principal components. The lines link the dots to the time axis, where the 1st principal component is decoded as a gray level bar. Note that the test scenario is not the same as in Fig. 4.2

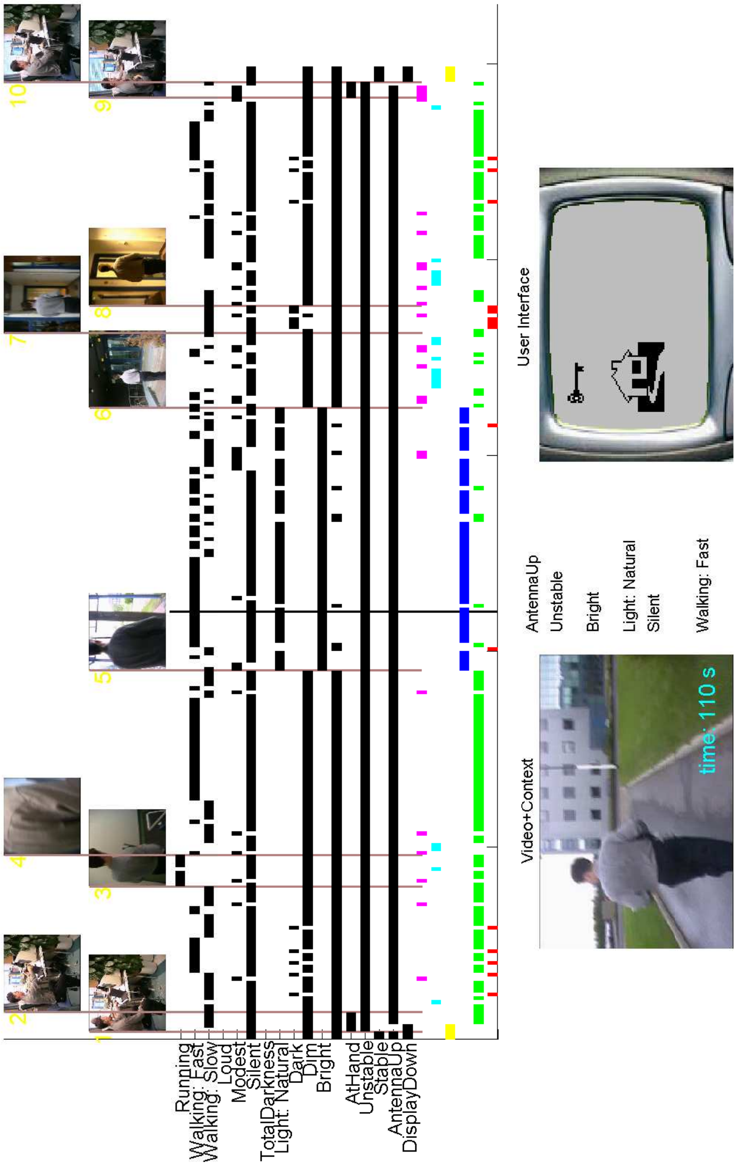


Figure 4.2: Time series segmentation to eleven minimum-variance segments and clustering to six clusters. The black horizontal bars show the activation of the context atoms. The thumbnail images with the vertical segment borders 1–10 show what happened at the segment borders (=putative context changes). The colored horizontal bars show into which of the six clusters the time instant belongs to. Certain UI profiles have been manually associated with each cluster. We know, e.g. that the cluster labeled with green is about walking outside while the yellow is about sitting in the office. Picture below the time series representation show the user at time t=110 s and the associated UI profile. (The black vertical line below thumbnail image 5 show the location of t=110 s in the time series.) In this case, the phone uses keypad lock and outdoors profile since the user is walking outside.

Chapter 5

Conclusions and future directions

In this thesis, I have developed data mining and visualization methods, to meet some of the practical needs in different applications involving multidimensional signals: industrial processes, biomedical signals, and sensor-based mobile-awareness and personalization. In particular, I have presented

- new ways to develop further the SOM as a data visualization method, e.g., to be used in process analysis,
- an exploratory method of investigating the stability of ICA estimates, e.g., in brain research,
- applications of data mining methods in mobile-awareness and personalization, i.e., using PCA, ICA and time-series segmentation in mobile-awareness, and a simple, adaptive algorithm in keyboard design, and
- enhancements and modifications of algorithms so that they would be better applicable to the data or problem at hand, i.e., the fast fixed-point algorithm for time series segmentation and a heuristic solution to the problem of finding a binary mixing matrix and independent binary sources.

The methods of (visual) data mining presented here are intended for assisting research and development. According to my experiences, rapid development and visualization also helps communicating and demonstrating ideas. This has been especially valuable in the research on context-awareness and personalization. Presenting even demonstrative implementation of ideas helps making decisions in projects, evaluating what the anticipated problems are, and claiming intellectual property rights. This is a communicative task for visualization: Bengtsson et al. (1996, 1997) study visualization as a means of facilitating communication and planning within engineering and other communities.

A wide, challenging and important topic within visual data mining would be to move on from creating techniques to comparing them with user tests on problems and techniques on a large scale. But meanwhile, there are a lot of immediate development steps to be taken, e.g, in the visual data mining tool *Icasso*; the implementations of clustering and visualization should be made more scalable. Furthermore, several datasets could be compared. In the near future, the toolbox will surely be developed in this direction.

In scope of context-awareness studies in this thesis, the approach described in (Flanagan et al., 2002) seems promising, and the next step is collecting a set of richer data, and most importantly, data on genuine user actions in order to get hands on any real applications. Another topic in this area would be to develop the adaptive keyboard further and test its impact on users in real applications.

I have motivated the research of context-awareness and personalization by using examples of triggered actions, e.g., inferring some everyday event and mapping a specific action to that. It is easy to make up such illustrative examples but, in reality, such functions are often not so easily implemented. Typical, rational examples of triggered actions, such as “silence the ring tone, if X”, are quite vulnerable to distracting false operation and unpredictable behavior. Moreover, I have left many aspects spanning from system architecture standardization to usability design.

Furthermore, the examples of automated, triggered actions may have hidden the fact that the emerging user interface paradigms are not just making existing services and applications easier to use. More generally, they create new types of potential applications, interaction modes, and social implications. I believe that jocularity, feel of personalization, social relations, lifestyle, games, and entertainment are at least as important boosters for developing personalization, context-awareness, and multimodality than making interaction more *effective*, i.e., “the device easier to use”. In fact, applications related to entertainment or lifestyle might also provide less failure-critical applications. There is probably a lot of potential in marketing, advertising and behavior surveillance for such techniques—with privacy issues intertwined. Analyzing context by measurements instead of user surveillance or diaries involving user input, can also be beneficial for usability research.

Socially important digital media, like WWW, news, and Internet chats, among others, are already very important application areas of data mining. It seems that the number of sensory and other streaming information from new type of personal and ubiquitous devices is steadily growing. In general, context recognition is a very challenging machine learning problem since it ultimately requires that a machine should have common-sense reasoning to react to everyday situations. Especially in mobile-aware applications, one must further consider the limited power and computation resources of a mobile device. Developing algorithms for efficient extraction, aggregation and analyze of relevant information from diverse sources is one of the key issues in this area. This surely presents new challenges for data mining as well as for machine learning in general.

Bibliography

Abello, J. and Korn, J. (2002). MGV: A System for Visualizing Massive Multidigraphs. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):21–38.

Aggarwal, C. C., Procopiuc, C., Wolf, J. L., Yu, P. S., and Park, J. S. (1999). Fast Algorithms for Projected Clustering. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 61–72, Philadelphia, PA.

Ainsworth, E. J. (1998). Classification of Ocean Colour Using Self-Organizing Feature Maps. In Yamakawa, T. and Matsumoto, G., editors, *Proceedings of the 5th International Conference on Soft Computing and Information/Intelligent Systems (IIZUKA98)*, volume 2, pages 996–999. World Scientific.

Alhoniemi, E. (2000). Analysis of Pulping Data Using the Self-Organizing Map. *Tappi Journal*, 83(7):66.

Alhoniemi, E. (2003). Simplified Time Series Representations for Efficient Analysis of Industrial Process Data. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 17(2):103–114.

Alhoniemi, E., Hollmén, J., Simula, O., and Vesanto, J. (1999). Process Monitoring and Modeling Using the Self-Organizing Map. *Integrated Computer-Aided Engineering*, 6(1):3–14.

Aristide, V. (1993). On the Use of Two Traditional Statistical Techniques to Improve the Readability of Kohonen Maps. In *Proceedings of NATO ASI workshop on Statistics and Neural Networks*.

Asimov, D. (1985). The Grand Tour: a Tool for Viewing Multidimensional Data. *SIAM Journal on Scientific and Statistical Computing*, 6(1):128–143.

Bandyopadhyay, S. and Maulik, U. (2001). Nonparametric Genetic Clustering: Comparison of Validity Indices. *IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews*, 31(1):120–125.

Basseville, M. (1989). Distance Measures for Signal Processing and Pattern Recognition. *Signal Processing*, 18(4):349–369.

Becker, R. A. and Cleveland, W. S. (1987). Brushing Scatterplots. *Technometrics*, 29(2):127–142.

Bellman, R. (1961). On the Approximation of Curves by Line Segments Using Dynamic Programming. *Communications of the ACM*, 4(6):284.

Bengtsson, P., Johansson, C., Eriksson, J., Johansson, G., af Klerckler, J., and Axelsson, K. (1996). Case study: Computer-Aided Planning of Production, Working and Residential Environments. *International Journal of Industrial Ergonomics*, 17:59–68.

Bengtsson, P., Johansson, C. R., and Akselsson, K. R. (1997). Planning Working Environment and Production by Using Paper Drawings and Computer Animation. *Ergonomics*, 40(3):334–347.

Bennett, K. D. (1996). Determination of the Number of Zones in a Biostratigraphical Sequence. *New Phytol.*, 132:155–170.

Bertin, J. (1983). *Semiology of Graphics*. The University of Wisconsin Press. (Translated by Berg, J.) Originally Bertin, J. (1967). Sémiologie graphique.

Bezdek, J. C. and Pal, N. R. (1998). Some New Indexes of Cluster Validity. *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, 28(3):301–315.

Bingham, E. (2003). Advances in Independent Component Analysis with Applications to Data Mining. D. Sc. (Tech.) Thesis. Report D4 in Series Dissertations in Computer and Information Science, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland.

Bingham, E., Mannila, H., and Seppänen, J. K. (2002). Topics in 0-1 Data. In Hand, D., Keim, D., and Ng, R., editors, *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 450–455, Edmonton, Alberta, Canada.

Bishop, C. M., Svensén, M., and Williams, C. K. I. (1998). The Generative Topographic Mapping. *Neural Computation*, 10(1):215–234.

Blake, C. L. and Merz, C. J. (1998). UCI Repository of Machine Learning Databases. http://www.ics.uci.edu/∼mlearn/MLRepository.html. [referred 25 Apr 2004].

Blom, J. O. and Monk, A. F. (2003). Theory of Personalization of Appearance: Why Users Personalize Their PCs and Mobile Phones. *Human–Computer Interaction*, 18:193–228.

Borg, I. and Groenen, P. (1997). *Modern Multidimensional Scaling. Theory and Applications*. Springer Series in Statistics. Springer.

Bottou, L. and Bengio, Y. (1995). Convergence Properties of the K-Means Algorithms. In Tesauro, G., Touretzky, D., and Leen, T., editors, *Neural Information Processing Systems 1994 (NIPS 1994)*, volume 7, pages 585–592. MIT Press.

Buja, A., McDonald, J. A., Michalack, J., and Stuetzle, W. (1991). Interactive Data Visualization Using Focusing and Linking. In *Proceedings of the IEEE Conference on Visualization (Visualization'91)*, pages 156–163, 419.

Cantoni, A. (1971). Optimal Curve Fitting with Piecewise Linear Functions. *IEEE Transactions on Computers*, C-20(1):59–67.

Card, S. K., Mackinlay, J. D., and Shneiderman, B. (1999a). Information Visualization. In Card et al. (1999b), chapter 1, pages 1–34.

Card, S. K., Mackinlay, J. D., and Shneiderman, B., editors (1999b). *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann.

Cardoso, J.-F. and Soloumiac, A. (1993). Blind Beamforming for Non-Gaussian Signals. *IEE-Proceedings-F*, 140(46):362–370.

Chen, C. and Yu, Y. (2000). Empirical Studies of Information Visualization: a Meta-Analysis. *International Journal of Human-Computer Studies*, 53(6):851–866.

Chen, G. and Kotz, D. (2000). A Survey of Context-Aware Mobile Computing Research. Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College.

Cherkassky, V. and Mulier, F. (1998). *Learning from Data: Concepts, Theory, and Methods*. John Wiley & Sons.

Chernoff, H. (1973). The Use of Faces to Represent Points in K-Dimensional Space Graphically. *Journal of the American Statistical Association*, 68(342):361–368.

Cheung, Y.-M. (2003). k^*-Means: A New Generalized k-Means Clustering Algorithm. *Pattern Recognition Letters*, 24(15):2883–2898.

Chi, E. H. (2000). A Taxonomy of Visualization Techniques Using the Data State Reference Model. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis 2000)*, pages 69–75.

Cios, K. J., Pedrycz, W., and Swiniarski, R. W. (1998). *Data Mining. Methods for Knowledge Discovery*. Kluwer Academic Publishers.

Clarkson, B., Mase, K., and Pentland, A. (2000). Recognizing User Context via Wearable Sensors. In *Digest of Papers of the Fourth International Symposium on Wearable Computers*, pages 69–75. IEEE.

Clarkson, B. and Pentland, A. (1999). Unsupervised Clustering of Ambulatory Audio and Video. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'99)*, volume 6, pages 3037–3040.

Cleveland, W. S. (1994). *The Elements of Graphing Data*. Hobart Press, New Jersey, Summit.

Cleveland, W. S. and McGill, R. (1984). The Many Faces of a Scatterplot. *Journal of the American Statistical Association*, 79(388):807–822.

Cleveland, W. S. and McGill, R. (1987). Graphical Perception: The Visual Decoding of Quantitative Information on Graphical Displays of Data. *Journal of the Royal Statistical Society. Series A (General)*, 150(3):192–229.

Comon, P. (1994). Independent Component Analysis—a New Concept? *Signal Processing*, (36):287–314.

Cormack, R. M. (1971). A Review of Classification. *Journal of the Royal Statistical Society. Series A (General)*, 134(3):321–367.

Davies, D. L. and Bouldin, D. W. (1979). A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227.

de Oliviera, M. C. F. and Levkowitz, H. (2003). From Visual Data Exploration to Visual Data Mining: A Survey. *IEEE Transactions on Visualization and Computer Graphics*, 9(3):378–394.

Demartines, P. and Hérault, J. (1993). Vector Quantization and Projection Neural Network. In Mira, J., Cabestany, J., and Prieto, A., editors, *International Workshop on Artificial Neural Networks*, volume 686 of *Lecture Notes in Computer Science*, pages 328–333. Springer Verlag.

Demartines, P. and Hérault, J. (1997). Curvilinear Component Analysis: A Self-Organizing Neural Network for Nonlinear Mapping of Data Sets. *IEEE Transactions on Neural Networks*, 8(1):148–154.

Dey, A. K. and Abowd, G. D. (1999). Towards a Better Understanding of Context and Context-Awareness. Technical Report GIT-GVU-99-22, GVU Center, Georgia Institute of Technology.

Dey, A. K., Abowd, G. D., and Salber, D. (2001). A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *Human–Computer Interaction*, 16(2–4):97–166.

Dourish, P. (2004). What We Talk About When We Talk About Context. *Personal and Ubiquitous Computing*, 8(1):19–30.

Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern Classification*. Wiley-Interscience, 2nd edition.

Dunn, J. C. (1974). Well Separated Clusters and Optimal Fuzzy Partitions. *Journal of Cybernetics*, 4:95–104.

Efron, B. and Tibshirami, R. J. (1998). *An Introduction to the Bootstrap*. Number 57 in Monographs on Statistics and Applied Probability. Chapman & Hall/CRC Press LLC, Reprint, originally published by Chapman & Hall, 1993 edition.

Eick, S. G. (1997). Engineering Perceptually Effective Visualizations for Abstract Data. In Nielson et al. (1997), chapter 8, pages 191–209.

Erwin, E., Obenmayer, K., and Schulten, K. (1992). Self-Organizing Maps: Ordering, Convergence Properties and Energy Functions. *Biological Cybernetics*, 67:47–55.

Everitt, B. S. (1993). *Cluster Analysis*. Arnold, 3rd edition.

Farringdon, J., Moore, A. J., Tillbury, N., Church, J., and Biemond, P. D. (1999). Wearable Sensor Badge and Sensor Jacket for Context Awareness. In *Digest of Papers of the Third International Symposium on Wearable Computers*, pages 45–51. IEEE.

Fayyad, U., Grinstein, G. G., and Wierse, A., editors (2002). *Information Visualization in Data Mining and Knowledge Discovery*. The Morgan Kaufman Series in Data Management Systems. Morgan Kaufman.

Fayyad, U. M., Piatetsky-Shapiro, G., and Smyth, P. (1996). From Data Mining to Knowledge Discovery: An Overview. In Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R., editors, *Advances in Knowledge Discovery and Data Mining*, chapter 1, pages 1–34. AAAI Press / The MIT Press.

Ferguson, E. S. (1977). The Mind's Eye: Nonverbal Thought in Technology. *Science*, 197(4306):827–836.

Fitts, P. M. (1954). The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement. *Journal of Experimental Psychology*, 47(6):381–391.

Flanagan, J. A. (2001). Topologically Ordered Neural Networks. In Moss, F. and Gielen, S., editors, *Neuro-Informatics and Neural Modelling*, volume 4 of *Handbook of Biological Physics*, chapter 16, pages 685–729. Elsevier.

Flanagan, J. A. (2003). A Non-Parametric Approach to Unsupervised Learning and Clustering of Symbol Strings and Sequences. In WSOM2003 (2003), pages 128–133. CD-ROM.

Flanagan, J. A., Mäntyjärvi, J., and Himberg, J. (2002). Unsupervised Clustering of Symbol Strings and Context Recognition. In ICDM2002 (2002), pages 171–178.

Fowlkes, E. B., Gnanadesikan, R., and Kettenring, J. R. (1988). Variable Selection in Clustering. *Journal of Classification*, 5:205–228.

Friedman, H. P. and Rubin, J. (1967). On Some Invariant Criteria for Grouping Data. *Journal of the American Statistical Association*, 62(320):1159–1178.

Friedman, J. H. and Tukey, J. W. (1974). A Projection Pursuit Algorithm for Exploratory Data Analysis. *IEEE Transactions on Computers*, C-23:881–890.

Gabriel, K. R. (1971). The Biplot Graphic Display of Matrices with Application to Principal Component Analysis. *Biometrika*, 58(3):453–467.

Gersho, A. and Gray, R. M. (1992). *Vector Quantization and Signal Compression*. Kluwer.

Gionis, A. and Mannila, H. (2003). Finding Recurrent Sources in Sequences. In *Proceedings of RECOMB'03*, Berlin. ACM.

Glymour, C., Madigan, D., Pregibon, D., and Smyth, P. (1997). Statistical Themes and Lessons for Data Mining. *Data Mining and Knowledge Discovery*, 1(1):11–28.

Golding, A. R. and Lesh, N. (1999). Indoor Navigation Using a Diverse Set of Cheap, Wearable Sensors. In *Digest of Papers of the Third International Symposium on Wearable Computers*, pages 29–36. IEEE.

Gordon, A. D. (1987). A Review of Hierarchical Classification. *Journal of the Royal Statistical Society. Series A (General)*, 150(2):119–137.

Gower, J. C. (1966). Some Distance Properties of Latent Root and Vector Methods Used in Multivariate Analysis. *Biometrika*, 53(3/4):325–338.

Gower, J. C. (1982). Measures of Similarity, Dissimilarity, and Distance. In Kotz, S. and Johnson, N. L., editors, *Encyclopedia of Statistical Sciences*, volume 5, pages 397–405. Wiley & Sons, New York.

Grabmeier, J. and Rudolph, A. (2002). Techniques of Cluster Algorithms in Data Mining. *Data Mining and Knowledge Discovery*, 6(4):303–360.

Greenberg, S. (2001). Context as a Dynamic Construct. *Human–Computer Interaction*, 16(2–4):257–268.

Grinstein, G. G., Hoffman, P. E., Pickett, R. M., and Laskowski, S. J. (2002). Benchmark Development for the Evaluation of Visualization for Data Mining. In Fayyad et al. (2002), chapter 11, pages 129–176.

Grinstein, G. G. and Ward, M. O. (2002). Introduction to Data Visualization. In Fayyad et al. (2002), chapter 1, pages 21–45.

Haase, H., Dai, F., Strassner, J., and Göbel, M. (1997). Immersive Investigation of Scientific Data. In Nielson et al. (1997), pages 35–58.

Halkidi, M., Batistakis, Y., and Vazirgiannis, M. (2001). On Clustering Validation Techniques. *Journal of Intelligent Information Systems*, 17(2/3):107–145.

Hand, D., Mannila, H., and Smyth, P. (2001). *Principles of Data Mining*. The MIT Press.

Healey, C. G., Booth, K. S., and Enns, J. T. (1996). High-Speed Visual Estimation Using Preattentive Processing. *ACM Transactions on Computer-Human Interaction*, 3(2):107–135.

Healey, C. G. and Enns, J. T. (1999). Large Datasets at a Glance: Combining Textures and Colors in Scientific Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 5(2):145–167.

Hearn, D. and Baker, M. P. (1997). *Computer Graphics, C Version*. Prentice Hall, 2nd edition.

Henderson, K. (1999). *On line and On Paper: Visual Representations, Visual Culture, and Computer Graphics in Design Engineering*. The MIT Press.

Hibino, S. L. (1999). A Task-Oriented View of Information Visualization. In *Proceedings of Computer-Human Interaction (CHI99)*, pages 178–179. ACM.

Himberg, J., Flanagan, J. A., and Mäntyjärvi, J. (2003). Towards Context Awareness Using Symbol Clustering Map. In WSOM2003 (2003), pages 249–254.

Hinneburg, A. and Keim, D. A. (1998). An Efficient Approach to Clustering in Large Multimedia Databases with Noise. In Agrawal, R. and Stolorz, P., editors, *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)*, pages 58–65, New York City, NY. ACM.

Hoffman, P., Grinstein, G., and Pinkey, D. (1999). Dimensional Anchors: A Graphic Primitive for Multidimensional Multivariate Information Visualizations. In *Proceedings of the 1999 Workshop on New Paradigms in Information Visualization and Manipulation (NPIVM '99)*, pages 9–16, Kansas City, Mo, USA.

Hoffman, P. E. and Grinstein, G. G. (2002). A Survey of Visualizations for High-Dimensional Data Mining. In Fayyad et al. (2002), chapter 2, pages 47–82.

Holtzblatt, K. (2003). Contextual Design. In Jacko and Sears (2003), chapter 49, pages 941–963.

Honkela, T., Kaski, S., Lagus, K., , and Kohonen, T. (1996). Exploration of Full-Text Databases with Self-Organizing Maps. In *Proceedings of IEEE International Conference on Neural Networks (ICNN'96)*, pages 56–61.

Horn, R. A. and Johnson, C. R. (1991). *Topics in Matrix Analysis*, chapter 5: The Hadamard product. Cambridge University Press.

Hsu, W. H., Welge, M., Redman, T., and Clutter, D. (2002). High-Performance Commercial Data Mining: A Multistrategy Machine Learning Application. *Data Mining and Knowledge Discovery*, 6(4):361–391.

Hyvärinen, A. (1999). Fast and Robust Fixed-Point Algorithms for Independent Component Analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634.

Hyvärinen, A., Karhunen, J., and Oja, E. (2001). *Independent Component Analysis*. Wiley Interscience.

Hyvärinen, A. and Oja, E. (1997). A Fast Fixed-Point Algorithm for Independent Component Analysis. *Neural Computation*, 9(7):1483–1492.

Hyvärinen, A. and Pajunen, P. (1999). Nonlinear Independent Component Analysis: Existence and Uniqueness Results. *Neural Networks*, 12(3):429–439.

ICDM2002 (2002). *Proceedings of the IEEE International Conference on Data Mining 2002 (ICDM2002)*, Maebashi, Japan. IEEE.

Iivarinen, J., Kohonen, T., Kangas, J., and Kaski, S. (1994). Visualizing the Clusters on the Self-Organizing Map. In Carlsson, C., Järvi, T., and Reponen, T., editors, *Proc. Conf. on Artificial Intelligence Res. in Finland*, number 12 in Conf. Proc. of Finnish Artificial Intelligence Society, pages 122–126, Helsinki, Finland. Finnish Artificial Intelligence Society.

Inselberg, A. and Dimsdale, B. (1990). Parallel Coordinates: A Tool for Visualizing Multidimensional Geometry. In *Proceedings of the First IEEE Conference on Visualization (Visualization'90)*, pages 361–378, Los Alamitos, California.

Jacko, J. A. and Sears, A., editors (2003). *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*. Human Factors and Ergonomics. Lawrence Erlbaum Associates.

Jain, A. K. and Dubes, R. C. (1988). *Algorithms for Clustering Data*. Prentice Hall.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data Clustering: A Review. *ACM Computing Surveys*, 31(3):264–323.

Jameson, A. (2003). Adaptive Interfaces and Agents. In Jacko and Sears (2003), chapter 15, pages 305–330.

Jutten, C. and Herault, J. (1991). Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, (24):1–10.

Kangas, J. A., Kohonen, T. K., and Laaksonen, J. T. (1990). Variants of Self-Organizing Maps. *IEEE Transactions on Neural Networks*, 1(1):93–99.

Kaski, S. (1997). *Data Exploration Using Self-Organizing Maps*. Number Ma 82 in Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series. The Finnish Academy of Technology, Helsinki University of Technology, Dept. of Computer Science, Espoo, Finland). D.Sc.(Tech.) Thesis.

Kaski, S., Kangas, J., and Kohonen, T. (1998a). Bibliography of Self-Organizing Map (SOM) Papers: 1981–1997. *Neural Computing Surveys*, 1:102–350. 3,343 references.

Kaski, S. and Lagus, K. (1996). Comparing Self-Organizing Maps. In *Proceedings of ICANN'96*, pages 809–814.

Kaski, S., Nikkilä, J., and Kohonen, T. (2000). Methods for Exploratory Cluster Analysis. In *Proceedings of International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet*, Scuola Superiore G. Reiss Romoli, L'Aquila, Italy. CD-ROM.

Kaski, S., Nikkilä, J., Oja, M., Venna, J., Törönen, P., and Castrén, E. (2003a). Trustworthiness and Metrics in Visualization of Gene Expression. *BMC Bioinformatics*, 4(48).

Kaski, S., Sinkkonen, J., and Klami, A. (2003b). Regularized Discriminative Clustering. In *IEEE International Workshop on Neural Networks for Signal Processing (NNSP2003)*, pages 289–298.

Kaski, S., Venna, J., and Kohonen, T. (1998b). Tips for Processing and Color-Coding of Self-Organizing Maps. In Deboeck, G. and Kohonen, T., editors, *Visual Explorations in Finance*, Springer Finance, chapter 14, pages 195–202. Springer-Verlag.

Kaski, S., Venna, J., and Kohonen, T. (1999). Coloring that Reveals High-Dimensional Structures in Data. In *Proceedings of the 6th International Conference on Neural Information Processing (ICONIP'99)*, volume 2, pages 729–734. IEEE.

Keim, D. A. (2002). Information Visualization and Visual Data Mining. *IEEE Transactions on Visualization and Computer Graphics*, 8:1–8.

Keim, D. A., Hao, M. C., and Dayal, U. (2002). Hierarchical Pixel Bar Charts. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):255–269.

Keim, D. A. and Kriegel, H.-P. (1996). Visualization Techniques for Mining Large Databases: A Comparison. *IEEE Transactions of Knowledge and Data Engineering*, 8(6):923–938.

Keogh, E., Chu, S., Hart, D., and Pazzani, M. (2001). An Online Algorithm for Segmenting Time Series. In *Proceedings of the First IEEE International Conference on Data Mining (ICDM 2001)*, pages 289–296, San José, California.

Kepner, J. and Kim, R. (2003). Cluster Detection in Databases: The Adaptive Matched Filter Algorithm and Implementation. *Data Mining and Knowledge Discovery*, 7(1):57–79.

Kim, W., Choi, B.-J., Hong, E.-K., Kim, S.-K., and Lee, D. (2003). A Taxonomy of Dirty Data. *Data Mining and Knowledge Discovery*, 7(1):81–99.

Kiviluoto, K. (1995). Topology Preservation in Self-Organizing Maps. Technical Report A29, Helsinki University of Technology, Laboratory of Computer and Information Science.

Kleinberg, J., Papadimitriou, C., and Raghavan, P. (1998). A Microeconomic View of Data Mining. *Data Mining and Knowledge Discovery*, 2(4):311–324.

Koffka, K. (1935). *Principles of Gestalt Psychology*. Lund Humphries.

Kohonen, T. (1982). Self-Organized Formation of Topologically Correct Feature Maps. *Biological Cybernetics*, 43(1):56–69.

Kohonen, T. (2001). *Self-Organizing Maps*. Springer, 3rd edition.

Kohonen, T., Hynninen, J., Kangas, J., and Laaksonen, J. (1996). SOM_PAK: The Self-Organizing Map Program Package. Technical Report A31, Helsinki University of Technology, Laboratory of Computer and Information Science.

Kohonen, T., Kaski, S., Lagus, K., Salojärvi, J., Honkela, J., Paatero, V., and Saarela, A. (2000). Self Organization of a Massive Document Collection. *IEEE Transactions on Neural Networks*, 11(3):574–585.

Kohonen, T. and Somervuo, P. (1998). Self-Organizing Maps of Symbol Strings. *Neurocomputing*, 21:19–30.

König, A. and Michel, T. (2003). DIPOL-SOM – A Distance Preserving Enhancement of the Self-Organizing Map for Dimensionality Reduction and Multivariate Data Visualization. In WSOM2003 (2003), pages 219–224. CD-ROM.

Korpipää, P., Koskinen, M., Peltola, J., Mäkelä, S.-M., and Seppänen, T. (2003). Bayesian Approach to Sensor-Based Context Awareness. *Personal and Ubiquitous Computing*, 7(2):113–124.

Kosslyn, S. M. (1994). *Elements of Graph Design*. W. H. Freeman and Company.

Laaksonen, J., Koskela, M., Laakso, S., and Oja, E. (2000). PicSOM – Content-Based Image Retrieval with Self-Organizing Maps. *Pattern Recognition Letters*, 21(13–14):1199–1207.

Laerhoven, K. V., Aidoo, K. A., and Lowette, S. (2001). Real-time analysis of data from many sensors with neural networks. In *Proceedings of the Fifth International Symposium on Wearable Computers*, pages 115–122, Zurich, Switzerland.

Laerhoven, K. V. and Cakmakci, O. (2000). What Shall We Teach Our Pants? In *Digest of Papers of the Fourth International Symposium on Wearable Computers*, pages 77–83. IEEE.

Laerhoven, K. V., Schmidt, A., and Gellersen, H.-W. (2002). Multi-Sensor Context Aware Clothing. In *Proceedings of the Sixth International Symposium on Wearable Computers(ISWC)*, pages 49–56, Seattle, Washington. IEEE.

Laine, S. (2003). Using Visualization, Variable Selection and Feature Extraction to Learn from Industrial Data. D. Sc. (Tech.) Thesis Report A69 in Publications in Computer and Information Science, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland.

Lampinen, J. and Kostiainen, T. (2002). Generative Probability Density Model in the Self-Organizing Map. In Seiffert and Jain (2002), chapter 4, pages 75–92.

Langley, P. (1994). Selection of relevant features in machine learning. In *Proceedings of the AAAI Fall Symposium on Relevance*, pages 140–144, New Orleans, LA. AAAI Press.

Lawrence, R. D., Almasi, G. S., and Rushmeier, H. E. (1999). A Scalable Parallel Algorithm for Self-Organizing Maps with Applications to Sparse Data Mining Problems. *Data Mining and Knowledge Discovery*, 3(2):171–195.

Leung, Y. K. and Apperley, M. D. (1994). A Review and Taxonomy of Distortion-Oriented Presentation Techniques. *ACM Transactions on Computer-Human Interaction*, 1(2):126–160.

Levine, E. and Domany, E. (2001). Resampling Method For Unsupervised Estimation of Cluster Validity. *Neural Computation*, 13(11):2573–2593.

Linde, Y., Buzo, A., and Gray, R. M. (1980). An Algorithm for Vector Quantizer Design. *IEEE Transactions on Communications*, COM-28(1):84–95.

Lohse, G. L. (1997). Models of Graphical Perception. In Helander, M., Landauer, T. K., and Prabu, P., editors, *Handbook of Human-Computer Interaction*, chapter 6, pages 107–135. Elsevier Science B.V., 2nd, completely revised edition.

MacKenzie, I. S. and Soukoreff, R. W. (2002). Text Entry for Mobile Computing: Models and Methods, Theory and Practice. *Human–Computer Interaction*, 17(2–3):147–198.

Mackinlay, J. D. (1986). Automating the Design of Graphical Presentations of Relational Information. *ACM Transactions on Graphics*, 5(2):111–141.

MacQueen, J. (1966). Some Methods for Classification and Analysis of Multivariate Observations. In Cam, L. M. L. and Neyman, J., editors, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume I, pages 281–297. University of California Press.

Maguire, M. (2001). Context of Use within Usability Activites. *International Journal of Human-Computer Studies*, 55:453–483.

Mäntyjärvi, J. (2003). Sensor-Based Context Recognition for Mobile Applications. Ph. D. Thesis, Oulu University, Faculty of Technology, Oulu, Finland VTT Publications 511, VTT Technical Research Centre of Finland, Espoo.

Mäntyjärvi, J., Himberg, J., Korpipää, P., and Mannila, H. (2001a). Extracting the Context of a Mobile Device User. In *Proceedings of the 8th IFAC/IFIP/IFORS/IEA International Symposium on Human-Machine Systems (HMS'01)*, pages 450–500, Kassel, Germany.

Mäntyjärvi, J., Himberg, J., and Seppänen, T. (2001b). Recognizing Human Motion with Multiple Acceleration Sensors. In *Proceedings of the 2001 IEEE International Conference on Systems, Man and Cybernetics (SMC2001)*, pages 747–752, Tucson, USA.

Mäntyjärvi, J., Huuskonen, P., and Himberg, J. (2002a). Collaborative Context Determination to Support Mobile Terminal Applications. *IEEE Wireless Communication*, 9(5):39–45.

Mäntyjärvi, J., Koivumäki, J., and Vuori, P. (2002b). Keystroke Recognition for Virtual Keyboards. In *Proceedings of the IEEE International Conference on Multimedia & Expo (ICME'02)*, pages 429–432, Naples, Italy.

Mäntyjärvi, J. and Seppänen, T. (2003). Adapting Applications in Handheld Devices Using Fuzzy Context Information. *Interacting with Computers*, 15(4):521–538.

Mäntyjärvi, J., Tuomela, U., Känsälä, I., and Häkkilä, J. (2003). Context Studio – Tool for Personalizing Context Aware Applications in Mobile Terminals. In *Proceedings of the Conference on the Computer-Human Interaction (OZCHI2003)*, University of Queensland, Brisbane, Australia. SIGCHI of Ergonomics Society of Australia.

Mäntylä, V.-M., Mäntyjärvi, J., Seppänen, T., and Tuulari, E. (2000). Hand Gesture Recognition of a Mobile Device User. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 281–284, NY, USA.

Maulik, U. and Bandyopadhay, S. (2002). Performance Evaluation of Some Clustering Algorithms and Validity Indices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(12):1650–1654.

Mead, A. (1992). Review of the Development of Multidimensional Scaling Methods. *The Statistician*, 41(1):27–39.

Meinecke, F., Ziehe, A., Kawanabe, M., and Müller, K.-R. (2002). A Resampling Approach to Estimate the Stability of One-Dimensional or Multidimensional Independent Components. *IEEE Transactions on Biomedical Engineering*, 49(12):1514–1525.

Milligan, G. W. (1989). A Validation Study of a Variable Weighting Algorithm for Cluster Analysis. *Journal of Classification*, 6(1):53–71.

Milligan, G. W. and Cooper, M. C. (1988). A study of Standardization of Variables in Cluster Analysis. *Journal of Classification*, 5(2):181–204.

Minsky, M. (2000). Commonsense-Based Interfaces. *Communications of the ACM*, 43(8):67–73.

Moran, T. P. and Dourish, P. (2001). Introduction to This Special Issue on Context-Aware Computing. *Human–Computer Interaction*, 16(2–4):87–95.

Muñoz, A. and Muruzábal, J. (1998). Self-Organizing Maps for Outlier Detection. *Neurocomputing*, 18(1–3):33–60.

Nielson, G. M., Hagen, H., and Müller, H., editors (1997). *Scientific Visualization*. IEEE Computer Society.

Nikkilä, J., Törönen, P., Kaski, S., Venna, J., Castrén, E., and Wong, G. (2002). Analysis and Visualization of Gene Expression Data Using Self-Organizing Maps. *Neural Networks*, 15:953–966.

Oates, T. (2002). PERUSE: An Unsupervised Algorithm for Finding Recurring Patterns in Time Series. In ICDM2002 (2002), pages 330–337.

Oja, E. (1982). A Simplified Neuron Model as a Principal Component Analyzer. *Journal of Mathematical Biology*, 15:267–273.

Oja, M., Kaski, S., and Kohonen, T. (2003). Bibliography of Self-Organizing Map (SOM) Papers: 1998-2001 Addendum. *Neural Computing Surveys*, pages 1–156. 2,092 references. Electronic journal at `http://www.cse.ucsc.edu/NCS/vol3.html` [referred 6 Oct 2004].

Oviatt, S. (2003). Multimodal Interfaces. In Jacko and Sears (2003), chapter 14, pages 286–304.

Pajunen, P. (1997). Blind Separation of Binary Sources with Less Sensors than Sources. In *Proceedings of the International Conference on Neural Networks (ICNN'97)*, volume 3, pages 1994–1997, Houston, Texas. IEEE, INNS.

Palmieri, F., Budillon, A., Calabrese, M., and Mattera, D. (1998). Searching for a Binary Factorial Code Using the ICA Framework. *Neurocomputing*, 22:131–144.

Pampalk, E., Goebl, W., and Widmer, G. (2003). Visualizing Changes in the Structure of Data for Exploratory Feature Selection. In *SIGKDD'03*, pages 157–166. ACM.

Pampalk, E., Rauber, A., and Merkl, D. (2002a). Content-Based Organization and Visualization of Music Archives. In *Proceedings of ACM Multimedia 2002*, pages 570–579, Juan-les-Pins, France.

Pampalk, E., Rauber, A., and Merkl, D. (2002b). Using Smoothed Data Histograms for Cluster Visualization in Self-Organizing Maps. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN 2002)*, pages 871–876, Madrid, Spain.

Partridge, K., Chatterjee, S., Sazawal, V., Borriello, G., and Want, R. (2002). TiltType: Accelerometer-Supported Text Entry for Very Small Devices. In *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology (UIST'02)*, pages 201–204, Paris, France.

Pascoe, J. (1998). Adding Generic Contextual Capabilities to Wearable Computers. In *Second International Symposium on Wearable Computers*, pages 92–99.

Patané, G. and Russo, M. (2001). The Enhanced LBG Algorithm. *Neural Networks*, 14:1219–1237.

Patel, P., Keogh, E., Lin, J., and Lonardi, S. (2002). Mining Motifs in Massive Time Series Databases. In ICDM2002 (2002), pages 370–377.

Pavlidis, T. (1973). Waveform Segmentation Through Functional Approximation. *IEEE Transactions on Computers*, C-22(7):689–697.

Pavlidis, T. (1980). Algorithms for Shape Analysis and Waveforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-2(4):301–312.

Peña, J., Lozano, J., and Larrañaga, P. (1999). An Empirical Comparison of Four Initialization Methods for the K-Means Algorithm. *Pattern Recognition Letters*, 20(10):1027–1040.

Polani, D. (2002). Measures for the Organization of Self-Organizing Maps. In Seiffert and Jain (2002), chapter 2, pages 13–44.

Prandoni, P., Goodwin, M., and Vetterli, M. (1997). Optimal Time Segmentation for Signal Modeling and Compression. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'97)*, volume 3, pages 2029–2032.

Provost, F. and Kolluri, V. (1999). A Survey of Methods for Scaling Up Inductive Algorithms. *Data Mining and Knowledge Discovery*, 3(2):131–169.

Pyle, D. (1999). *Data Preparation for Data Mining*. Morgan Kaufmann Publishers.

Ritter, H. (1999). Self-Organizing Maps on Non-Euclidean Spaces. In Oja, E. and Kaski, S., editors, *Kohonen Maps*, pages 97–109. Elsevier Science.

Roweis, S. and Saul, L. (2000). Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500):2323–2325.

Rubio, M. and Giménez, V. (2003). New Methods for Self-Organizing Map Visual Analysis. *Neural Computation and Applications*, 12:142–152.

Salam, F. M. and Erten, G. (1999). Sensor Fusion by Principal and Independent Component Decomposition Using Neural Networks. In *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 211–215, Taipei, Taiwan, R.O.C.

Sammon, Jr., J. W. (1969). A Nonlinear Mapping for Data Structure Analysis. *IEEE Transactions on Computers*, C-18(5):401–409.

Schalkoff, R. (1992). *Pattern Recognition: Statistical, Structural and Neural Approaches*. John Wiley & Sons, Inc.

Schilit, B. N., Adams, N., and Want, R. (1994). Context-Aware Computing Applications. In *Proceedings of the Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA. IEEE Computer Society.

Schmidt, A., Aidoo, K. A., Takaluoma, A., Tuomela, U., Laerhoven, K. V., and de Velde, W. V. (1999a). Advanced Interaction in Context. In *Proceedings of First International Symposium on Handheld and Ubiquitous Computing (HUC99)*, Lecture Notes on Computer Science 1707, pages 89–101, Karlsruhe, Germany. Springer-Verlag.

Schmidt, A., Beigl, M., and Gellersen, H.-W. (1999b). Sensor-Based Adaptive Mobile User Interfaces. In *Proceedings of the 8th International Conference on Human-Computer Interaction*, volume 2, pages 251–255.

Schmidt, A., Takaluoma, A., and Mäntyjärvi, J. (2000). Context Aware Telephony Over WAP. *Personal Technologies*, 4(4):225–229.

Sebrechts, M. M., Vasilakis, J., Miller, M. S., Cugini, J. V., and Laskowski, S. J. (1999). Visualization of Search Results: A Comparative Evaluation of Text, 2D, and 3D Interfaces. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–10, Berkeley, California, United States.

Seiffert, U. and Jain, L. C., editors (2002). *Self-Organizing Neural Networks: Recent Advances and Applications*. Studies in Fuzziness and Soft Computing. Physica-Verlag.

Seppänen, J. K., Bingham, E., and Mannila, H. (2003). A Simple Algorithm for Topic Identification in 0-1 Data. In *Knowledge Discovery in Databases: PKDD 20003. 7th European Conference on Principles and Practice of Knowledge Discovery in Databases*, number 2838 in Lecture Notes in Artificial Intelligence, pages 423–434, Cavtat-Dubrovnik, Croatia. Springer.

Shatkay, H. and Zdonik, S. B. (1996). Approximate Queries and Representations for Large Data Sequences. In *Proceedings of the 12th International Conference on Data Engineering*, pages 536–545. IEEE.

Shearer, C. (2000). The CRISP-DM Model: The New Blueprint for Data Mining. *Journal of Data Warehousing*, 5(4):13–22.

Sibson, R. (1979). Studies in the Robustness of Multidimensional Scaling: Perturbational Analysis of Classical Scaling. *Journal of the Royal Statistical Society. Series B (Methodological)*, 41(2):217–229.

Silfverberg, M., MacKenzie, I. S., and Korhonen, P. (2000). Predicting Text Entry on Mobile Phones. In *Proceedings of Computer Human Interaction (CHI'2000)*, pages 9–16, The Hague, The Netherlands. ACM.

Simula, O., Alhoniemi, E., Hollmén, J., and Vesanto, J. (1996). Monitoring and Modeling of Complex Processes Using Hierarchical Self-Organizing Maps. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS'96)*, volume Supplement, pages 73–76.

Sinkkonen, J. (2003). Learning Metrics and Discriminative Clustering. D. Sc. (Tech.) Thesis. Report D2 in Series Dissertations in Computer and Information Science, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland.

Sinkkonen, J. and Kaski, S. (2002). Clustering Based on Conditional Distributions in an Auxiliary Space. *Neural Computation*, 14:217–239.

Skupin, A. (2002). A Cartographic Approach to Visualizing Conference Abstracts. *IEEE Computer Graphics and Applications*, 22(1):50–58.

Stolte, C., Tang, D., and Hanrah, P. (2002). Polaris: A System for Query, Analysis, and Visualization of Multidimensional Relational Databases. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):52–65.

Swayne, D. F., Lang, D. T., Buja, A., and Cook, D. (2003). GGobi: Evolving from XGobi into an Extensible Framework for Interactive Data Visualization. *Computational Statistics & Data Analysis*, 43:423–444.

Taleb, A. and Jutten, C. (1999). Source Separation in Post-Nonlinear Mixtures. *IEEE Transactions on Signal Processing*, 47(10):2807–2820.

Tenenbaum, J., de Silva, V., and Langford, J. (2000). A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323.

Tomasi, C., Rafii, A., and Torunogolu, I. (2003). Full-Size Projection Keyboard for Handheld Devices. *Communications of the ACM*, 46(7):71–75.

Tryba, V., Metzen, S., and Goser, K. (1989). Designing of Basic Integrated Circuits by Self-Organizing Feature Maps. In *International Workshop on Neural Networks and their Applications*, pages 225–235, Nanterre, France.

Tufte, E. R. (1983). *The Visual Display of Quantitative Information*. Graphics Press.

Tufte, E. R. (1990). *Envisioning Information*. Graphics Press.

Tukey, J. W. (1977). *Exploratory Data Analysis*. Addison-Wesley Series in Behavioral Science: Quantitative Methods. Addison-Wesley.

Ultsch, A. (2003). Maps for the Visualization of High-Dimensional Data Spaces. In WSOM2003 (2003). CD-ROM.

Ultsch, A. and Siemon, H. P. (1990). Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis. In *Proc. INNC'90, Int. Neural Network Conf.*, pages 305–308, Dordrecht, Netherlands. Kluwer.

van Liere, R. and de Leeuw, W. (2003). GraphSplatting: Visualizing Graphs as Continuous Fields. *IEEE Transactions on Visualization and Computer Graphics*, 9(2):206–212.

van Walsum, T., Post, F. H., Silver, D., and Post, F. J. (1996). Feature Extraction and Iconic Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 2(2):111–119.

Vasko, K. and Toivonen, H. T. (2002). Estimating the Number of Segments in Time Series Data Using Permutation Tests. In ICDM2002 (2002), pages 466–473.

Venna, J. and Kaski, S. (2001). Neighborhood Preservation in Nonlinear Projection Methods: An Experimental Study. In Dorffner, G., Bischof, H., and Hornik, K., editors, *Artificial Neural Networks (ICANN 2001)*, pages 485–491. Springer.

Vesanto, J. (1999). SOM-Based Data Visualization Methods. *Intelligent Data Analysis*, 3(2):111–126.

Vesanto, J. (2000). Neural Network Tool for Data Mining: SOM Toolbox. In *Proceedings of the Symposium on Tool Environments and Development Methods for Intelligent Systems (TOOLMET2000)*, pages 184–196, Oulu, Finland. Oulun yliopistopaino.

Vesanto, J. (2002). *Data Exploration Process Based on the Self-Organizing Map*. Number Ma 115 in Acta Polytechnica Scandinavica, Mathematics and Computing Series. Finnish Academies of Technology, Helsinki University of Technology, Espoo, Finland. Ph. D. Thesis.

Vesanto, J. and Ahola, J. (1999). Hunting for Correlations in Data Using the Self-Organizing Map. In Bothe, H., Oja, E., Massad, E., and Haefke, C., editors, *Proceeding of the International ICSC Congress on Computational Intelligence Methods and Applications (CIMA'99)*, pages 279–285. ICSC Academic Press.

Vesanto, J., Alhoniemi, E., Himberg, J., Kiviluoto, K., and Parviainen, J. (1999). Self-Organizing Map for Data Mining in MATLAB: the SOM Toolbox. *Simulation News Europe*, (25):54.

Vesanto, J., Himberg, J., Alhoniemi, E., and Parhankangas, J. (2000). SOM Toolbox for Matlab 5. Report A57, Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland.

Vesanto, J. and Hollmén, J. (2002). An Automated Report Generation Tool for the Data Understanding Phase. In Ajith Abraham, A. and Köppen, M., editors, *Hybrid Information Systems*, Advances in Soft Computing, pages 611–626. Physica Verlag. based on articles in Int. Workshop on Hybrid Intelligent Systems (HIS'01) in conjunction with the Australian Joint Conf. on Artificial Intelligence (AI'01). December 11–12, 2001, Adelaide, Australia.

Vuori, V. and Laaksonen, J. (2002). A Comparison of Techniques for Automatic Clustering of Handwritten Characters. In *Proceedings of the 16th International Conference on Pattern Recognition*, volume 3, pages 168–171, Quebec, Canada. IEEE.

Vuori, V., Laaksonen, J., Oja, E., and Kangas, J. (2000). Controlling On-Line Adaptation of a Prototype-Based Classifier for Handwritten Characters. In *Proceedings the of 15th International Conference on Pattern Recognition*, volume 2, pages 331–334. IEEE.

Wang, C. and Wang, X. S. (2000). Supporting Content-based Searches on Time Series via Approximation. In *Proceedings of the 12th International Conference on Scientific and Statistical Database Management*, pages 69–81. IEEE.

Want, R., Hopper, A., Falcao, V., and Title, J. G. (1992). The Active Badge Location System. *ACM Transactions on Information Systems*, 10(1):91–102.

Ward, D. J., Blackwell, A. F., and MacKay, D. J. (2002). Dasher: A Gesture-Driven Data Entry Interface for Mobile Computing. *Human–Computer Interaction*, 17:199–228.

Ware, C. (1988). Color Sequences for Univariate Maps: Theory, Experiments, and Principles. *IEEE Computer Graphics and Applications*, 8(5):41–49.

Ware, C. (2000). *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers.

Ware, C. and Beatty, J. C. (1988). Using Color Dimensions to Display Data Dimensions. *Human Factors*, 30(2):127–142.

Weiser, M. (1991). The Computer for the 21st Century. *Scientific American*, 265(3):66–75.

Wigdor, D. and Balakrishnan, R. (2003). TiltText: Using Tilt for Text Input to Mobile Phones. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology (UIST'03)*, pages 81–90, Vancouver, Canada.

Wong, P. C. (1999). Visual Data Mining. *IEEE Computer Graphics and Applications*, 19(5):20–21.

Wong, P. C. and Bergeron, R. D. (1997). 30 Years of Multidimensional Multivariate Visualization. In Nielson et al. (1997), pages 3–33.

WSOM2003 (2003). *Proceedings of the Workshop on Self-Organizing Maps (WSOM2003)*, Hibino, Kitakyushu, Japan.

Xiong, Z., Herly, C., Ramchandran, K., and Orchard, M. T. (1994). Flexible Time Segmentations for Time-Varying Wavelet Packets. In *IEEE Proc. Intl. Symp on Time-Frequency and Time-Scale Analysis*, pages 9–12.

Yang, C. C., Chen, H., and Hong, K. (2003). Visualization of Large Category Map for Internet Browsing. *Decision Support Systems*, 35:89–102.

Yin, H. (2001). Visualization Induced SOM (ViSOM). In Allinson, N., Yin, H., Allinson, L., and Slack, J., editors, *Advances in Self-Organizing Maps*, pages 81–88. Springer.

Zhai, S., Hunter, M., and Smith, B. A. (2002). Performance Optimization of Virtual Keyboards. *Human–Computer Interaction*, 17:229–269.

Zhang, J. (1996). A Representational Analysis of Relational Information Displays. *International Journal of Human–Computer Studies*, 45:59–74.

Zhang, J. (1997). The Nature of External Representations in Problem Solving. *Cognitive Science*, 21(2):179–217.

Zhang, T., Ramakrishnan, R., and Livny, M. (1996). BIRCH: An Efficient Data Clustering Method for Very Large Databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pages 103–114.