Ma 82 UDC 681.327.12:159.95:519.2

ACTA
POLYTECHNICA
SCANDINAVICA

MATHEMATICS, COMPUTING AND MANAGEMENT IN ENGINEERING
SERIES No. 82

Data Exploration Using Self-Organizing Maps

SAMUEL KASKI

Helsinki University of Technology
Neural Networks Research Centre
Rakentajanaukio 2C
FIN-02150 Espoo, Finland

Thesis for the degree of Doctor of Technology to be presented with due permission for public examination and criticism in Auditorium F1 of the Helsinki University of Technology on the 21st of March, at 12 o'clock noon.

Helsinki University of Technology
Department of Computer Science and Engineering
Laboratory of Computer and Information Science

ESPOO 1997

Kaski, S., **Data exploration using self-organizing maps**. Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series No. 82, Espoo 1997, 57 pp. Published by the Finnish Academy of Technology. ISBN 952-5148-13-0. ISSN 1238-9803. UDC 681.327.12:159.95:519.2

Keywords: Data mining, exploratory data analysis, multivariate analysis, neural networks, self-organizing map, SOM

ABSTRACT

Finding structures in vast multidimensional data sets, be they measurement data, statistics, or textual documents, is difficult and time-consuming. Interesting, novel relations between the data items may be hidden in the data. The self-organizing map (SOM) algorithm of Kohonen can be used to aid the exploration: the structures in the data sets can be illustrated on special map displays.

In this work, the methodology of using SOMs for exploratory data analysis or data mining is reviewed and developed further. The properties of the maps are compared with the properties of related methods intended for visualizing high-dimensional multivariate data sets. In a set of case studies the SOM algorithm is applied to analyzing electroencephalograms, to illustrating structures of the standard of living in the world, and to organizing full-text document collections.

Measures are proposed for evaluating the quality of different types of maps in representing a given data set, and for measuring the robustness of the illustrations the maps produce. The same measures may also be used for comparing the knowledge that different maps represent.

Feature extraction must in general be tailored to the application, as is done in the case studies. There exists, however, an algorithm called the adaptive-subspace self-organizing map, recently developed by Kohonen, which may be of help. It extracts invariant features automatically from a data set. The algorithm is here characterized in terms of an objective function, and demonstrated to be able to identify input patterns subject to different transformations. Moreover, it could also aid in feature exploration: the kernels that the algorithm creates to achieve invariance can be illustrated on map displays similar to those that are used for illustrating the data sets.

PREFACE

This work has been carried out in the Neural Networks Research Centre, Helsinki University of Technology, during 1993-1996. While completing the thesis I have participated in several projects of the centre, under the supervision of Academy Professor Teuvo Kohonen. I wish to thank him for guidance and for providing the facilities and financial support necessary for the studies. Most important, however, have been the numerous discussions that have had a fundamental effect on the work, not to mention the contributions of Professor Kohonen in the projects themselves and in co-authoring the majority of the presented publications.

I also wish to thank the other co-workers in the projects that have been included in the thesis, Timo Honkela, Harri Lappalainen, Sirkka-Liisa Joutsiniemi, and Krista Lagus, and also the rest of the personnel of the Neural Networks Research Centre and the Laboratory of Computer and Information Science. Special thanks for comments on the manuscript are due to Timo Honkela, Krista Lagus, Janne Sinkkonen, and Professors Teuvo Kohonen and Erkki Oja.

Numerous other people with whom I have worked and lived in other projects, some of which are hopefully lifelong, deserve many thanks as well. Those acknowledgments relate to some completely different stories, however.

The financial support of the Academy of Finland and the Jenny and Antti Wihuri Foundation is gratefully acknowledged.

Espoo, February 1997

Samuel Kaski

CONTENTS

LIST OF PUBLICATIONS

The thesis consists of an introduction and the following publications:

1. Joutsiniemi, S.-L., Kaski, S., and Larsen, T. A. (1995) Self-organizing map in recognition of topographic patterns of EEG spectra. *IEEE Transactions on Biomedical Engineering*, 42:1062–1068.

2. Kaski, S. and Kohonen, T. (1996) Exploratory data analysis by the Self-organizing map: Structures of welfare and poverty in the world. In Refenes, A.-P. N., Abu-Mostafa, Y., Moody, J., and Weigend, A., editors, *Neural Networks in Financial Engineering. Proceedings of the Third International Conference on Neural Networks in the Capital Markets*, pages 498–507. World Scientific, Singapore.

3. Kaski, S., Honkela, T., Lagus, K., and Kohonen, T. (1996) Creating an order in digital libraries with self-organizing maps. In *Proceedings of WCNN'96, World Congress on Neural Networks*, pages 814–817. Lawrence Erlbaum and INNS Press, Mahwah, NJ.

4. Kohonen, T., Kaski, S., Lagus, K., and Honkela, T. (1996) Very large two-level SOM for the browsing of newsgroups. In von der Malsburg, C., von Seelen, W., Vorbrüggen, J. C., and Sendhoff, B., editors, *Proceedings of ICANN'96, International Conference on Artificial Neural Networks*, Lecture Notes in Computer Science, vol. 1112, pages 269–274. Springer, Berlin.

5. Lagus, K., Honkela, T., Kaski, S., and Kohonen, T. (1996) Self-organizing maps of document collections: a new approach to interactive exploration. In Simoudis, E., Han, J., and Fayyad, U., editors, *Proceedings of the Second International Conference on Knowledge Discovery & Data Mining*, pages 238–243. AAAI Press, Menlo Park, CA.

6. Kaski, S. (1997) Computationally efficient approximation of a probabilistic model for document representation in the WEBSOM full-text analysis method. Accepted to *Neural Processing Letters.*

7. Kaski, S. and Lagus, K. (1996) Comparing self-organizing maps. In von der Malsburg, C., von Seelen, W., Vorbrüggen, J. C., and Sendhoff, B., editors, *Proceedings of ICANN'96, International Conference on Artificial Neural Networks*, Lecture Notes in Computer Science, vol. 1112, pages 809–814. Springer, Berlin.

8. Kohonen, T., Kaski, S., and Lappalainen, H. Self-organized formation of various invariant-feature filters in the Adaptive-Subspace SOM. Accepted to *Neural Computation.*

THE AUTHOR'S CONTRIBUTION

Publication 1 deals with the monitoring and exploration of EEG waveforms. Here the other authors were primarily responsible for the medical aspects and implications of the study. The methodology related to preprocessing, computation of the maps, and visualization of the results was mainly the responsibility of the author of the thesis.

In Publication 2 the idea of computing the taxonomies of countries with the SOM based on socioeconomic data, as well as valuable guidance and contributions to the writing of the article are credited to Professor Teuvo Kohonen. The author of the thesis designed and conducted the practical part of the study.

Studies on the WEBSOM full-text document analysis method are described in Publications 3, 4, 5, and 6. The basic system is presented in Publication 3, subsequently developed vast maps in Publication 4, and the exploration interface and suggested ways of using it in Publication 5. The original idea of using a two-stage SOM architecture for organizing document collections was due to Mr. Timo Honkela, and the methods for producing huge SOMs efficiently (presented in Publication 4) were originated by Professor Kohonen. Most other ideas and details, and the implementation and the experiments were developed jointly as a team, and it is not possible to give a full account of all of the differences in the contributions of the team members. The ideas that are predominantly due to the author of the thesis are the efficient encoding of the documents by table lookups and a subsequent convolution, analyzed in detail in Publication 6, and the entropy-based weighting of the words in computing word category histograms (described in Publication 4).

New procedures for comparing self-organized maps were introduced in Publication 7. The realization of the potential importance of comparing maps without judging their relative goodness was due to Ms Krista Lagus. Professor Kohonen should also be credited for valuable discussions related to comparing SOMs. The idea of the presented goodness measure as well as the design of the measures and the experiments was predominantly due to the author of the thesis.

In Publication 8 the ASSOM architecture conceived by Professor Kohonen (Kohonen, 1995a; Kohonen, 1995b; Kohonen, 1995c; Kohonen, 1996) is described, and a set of experiments made to demonstrate its capabilities is reported in detail. The theoretical aspects of the ASSOM, some minor details notwithstanding, are due to Professor Kohonen. The contributions of the author of the thesis are the joint development of the mathematical analysis part of the Publication with Professor Kohonen, and the joint conduction of the experiments with Mr. Harri Lappalainen.

LIST OF SYMBOLS AND ABBREVIATIONS

ASSOM	Adaptive-subspace self-organizing map
DSS	Decision support system
EEG	Electroencephalogram
EM	Expectation maximization
GNP	Gross national product
GTM	Generative topographic mapping
IR	Information retrieval
KDD	Knowledge discovery in databases
MDS	Multidimensional scaling
PCA	Principal component analysis
RBF	Radial basis function
SOM	Self-organizing map

$\mathbf{x}$, $\mathbf{x}_k$	input vector (data item), kth input vector
t	discrete time index
N	number of input vectors
$\mathbb{R}^n$	input space; n-dimensional Euclidean space
$\mathbf{m}_i$	ith cluster centroid, ith model vector
$c = c(\mathbf{x})$	index of the centroid (or model vector) that is closest to $\mathbf{x}$
K	number of cluster centroids (and reference vectors)
$\mathbf{x}'_k$	projection of $\mathbf{x}_k$
$d(k,l)$	distance between $\mathbf{x}_k$ and $\mathbf{x}_l$
$d'(k,l)$	distance between $\mathbf{x}'_k$ and $\mathbf{x}'_l$
E_M	cost function of the metric MDS method
E_N	cost function of the nonmetric MDS method
f	a monotonically increasing function used in nonmetric MDS
E_S	cost function of Sammon's mapping
q	inherent dimensionality of the data
h_{ij}	neighborhood kernel in the SOM algorithm
$\mathbf{r}_i$	location of the ith map unit on the map grid
$p(\mathbf{x})$	probability density function of $\mathbf{x}$
V_k	Voronoi-region corresponding to $\mathbf{m}_k$, viz. the set consisting of those $\mathbf{x}$ for which $c(\mathbf{x}) = i$
N_i	number of data items in V_k
$\mathbf{n}_i = 1/N_i \sum_{\mathbf{x}_k \in V_i} \mathbf{x}_k$	centroid of V_i
$\mathcal{O}$	computational complexity ("of the order of")
$E_{M'}$	a modified cost function of the metric MDS method
F	a decreasing weighting function

1 INTRODUCTION

It is relatively easy to give answers to well-specified questions about the statistical nature of a well-understood data set, like "how large should an aeroplane cockpit be to accommodate any one of the 95% of the potential pilots?" In this example the data may be assumed to be normally distributed and it is straightforward to estimate the threshold. The more data there is available the more accurate an answer can be given.

If, on the other hand, the data is not well-understood and the problem is not well-specified, an increase in the amount of data may even have the opposite effect. This holds for multivariate data in particular. If the goal is simply to try to make sense out of a data set to generate sensible hypotheses or to find some interesting novel patterns, it paradoxically seems that the more data there is available the more difficult it is to understand the data set. The structures are hidden among the large amounts of multivariate data. When exploring a data set for new insights, only methods that *discover and illustrate effectively the structures in the data* can be of help. Such methods, applied to large data sets, are the topic of this work.

A data-driven search for statistical insights and models is traditionally called *exploratory data analysis* (Hoaglin, 1982; Jain and Dubes, 1988; Tukey, 1977; Velleman and Hoaglin, 1981) in statistical literature. The process of making statistical inferences often consists of an exploratory, data-driven phase, followed by a confirmatory phase in which the reproducibility of the results is investigated. There thus exists a wealth of applications in which data sets need to be summarized to gain insight into them; the goal in this work is to present a data set in a form that is easily understandable but that at the same time preserves as much of the essential information in the data as possible.

Exploratory data analysis methods can be used as tools in knowledge discovery in databases (KDD) (Fayyad, 1996; Fayyad et al., 1996a; Fayyad et al., 1996c; Simoudis, 1996). In this relatively recently established field the emphasis is on the whole interactive process of knowledge discovery, discovery of novel patterns or structures in the data. The process consists of a multitude of steps starting from setting up the goals to evaluating the results, and possibly reformulating the goals based on the results. *Data mining*[1] is one step in the discovery process, a step in which suitable tools from many other disciplines including exploratory data analysis are used to find interesting patterns in the data. Depending on the goals of the data mining process essentially any kinds of pattern recognition (Devijver and Kittler, 1982; Fu, 1974; Fukunaga, 1972; Therrien, 1989; Schalkoff, 1992), machine learning (Forsyth, 1989; Langley, 1996; Michalski, 1983), and multivariate analysis (Cooley and Lohnes, 1971; Hair, Jr. et al., 1984; Kendall,

[1]Not all authors use the same terminology; *data mining* can also be used to refer to whole process, even as a synonym of KDD.

1975) algorithms may be useful; for recent examples, cf. Fayyad et al. (1996b). An essential novelty in the field then lies in emphasizing the discovery of previously unknown structures from vast databases, and in emphasizing the importance of considering the whole process.

In this work exploratory data analysis methods which illustrate the structures in data sets, are applied to large databases. The tool in this endeavor will be the self-organizing map (SOM) (Kohonen, 1982; Kohonen, 1995c). Some properties that distinguish the SOM from the other data mining tools are that it is numerical instead of symbolic, nonparametric, and capable of learning without supervision. The numerical nature of the method enables it to treat numerical statistical data naturally, and to represent graded relationships. Because the method does not require supervision and is nonparametric, used here in the sense that no assumptions about the distribution of the data need to be made, it may even find quite unexpected structures from the data.

In this thesis the relation of the SOM to some other data visualization and clustering methods is first analyzed in Section 2. Then, recipes on how to use the SOM in exploratory data analysis are given in Section 3. The areas of application that are treated in the Publications are introduced in Section 4, and finally two recent developments in the methodology are discussed in Section 5.

2 METHODS FOR EXPLORATORY DATA ANALYSIS

There exist several methods for quickly producing and visualizing simple summaries of data sets (Tukey, 1977). For example, the so-called five-number summary consisting of the smallest and largest data value, the median, and the first and third quartiles can be visualized as a drawing, where each number corresponds to some constituent like the altitude of a box.

Such simple methods are very useful for summarizing low-dimensional data sets, but as the dimensionality increases their ability to visualize interdimensional relations soon degrades.

In this section some methods for illustrating *structures*, multivariate relations between data items, in high-dimensional data sets will be discussed. The treatment will be restricted to methods that regard the inputs as metric vectors and that can be used without making assumptions about the distribution of the data. It is also assumed that no external information like class labels is available on the data items. The illustrations will then be driven solely by the actual structures in the data and not by prespecified assumptions about the class structure. Although the analysis is unsupervised, the possible class labels may be used *afterwards* to aid in the interpretation of the results; then they do not affect the structures that have been found.

The vectors in the input data set will be denoted by $\mathbf{x}_k$, $k = 1, \ldots, N$. Here $\mathbf{x}_k \in \mathbb{R}^n$. In statistics it is customary to call the components of the data vectors *observations* recorded on *variables*. Here the mathematical terminology will be preferred, however. The components may also be called *features* as is customary in pattern recognition literature.

It this section the emphasis will be on methods that illustrate structures in given, prespecified data sets. It may be useful to note, however, that in practical applications the *selection* and *preprocessing* of the data may be even more important than the choice of the analysis method. For example, changes in the relative scales of the features have a drastic effect on the results of most of the methods that will be presented: the larger the scale of a component the more the component affects the result. It is, however, very difficult to give general guidelines for the very application specific task of preprocessing; the approaches used in some case studies will be discussed in Section 3.1.

The following questions play a central role in applying a method to large, high-dimensional data sets: what kinds of structures the method is able to extract from the data set, how does it illustrate the structures, does it reduce the dimensionality of the data, and does it reduce the number of data items.

2.1 Visualization of high-dimensional data items

Several graphical means have been proposed for visualizing high-dimensional data items directly, by letting each dimension govern some aspect of the visualization and then integrating the results into one figure (cf., e.g., du Toit et al., 1986; Jain and Dubes, 1988). These methods can be used to visualize any kinds of high-dimensional data vectors, either the data items themselves or vectors formed of some descriptors of the data set like the five-number summaries (Tukey, 1977).

Perhaps the simplest method to visualize a data set is to plot a "profile" of each item, i.e., a two-dimensional graph in which the dimensions are enumerated on the x-axis and the corresponding values on y (Fig. 1 **a**). An alternative is a scatterplot where two original dimensions of the data are chosen to be portrayed as the location of an icon, and the rest of the dimensions are depicted as properties of the icon. For example the lengths of rays emanating from the center of the icon may visualize the values of the rest of the components (Fig. 1 **b**). Also the familiar pie diagrams can be used.

Andrews' curves (Andrews, 1972), one curve for each data item, are obtained by using the components of the data vectors as coefficients of orthogonal sinusoids, which are then added together pointwise (Fig. 1 **c**).

Chernoff's faces (Chernoff, 1973) are among the most famous visual displays. Each dimension of the data determines the size, location, or shape of some component of a facial caricature (Fig. 1 **d**). For example, one component is associated with the width of the mouth, another with the separation of the eyes, etc.

The major drawback that applies to all these methods in the data mining setting is that they do not, used as such, reduce the amount of data. If the data set is large, the display consisting of all the data items portrayed separately will be incomprehensible. The methods could, however, be useful for illustrating some

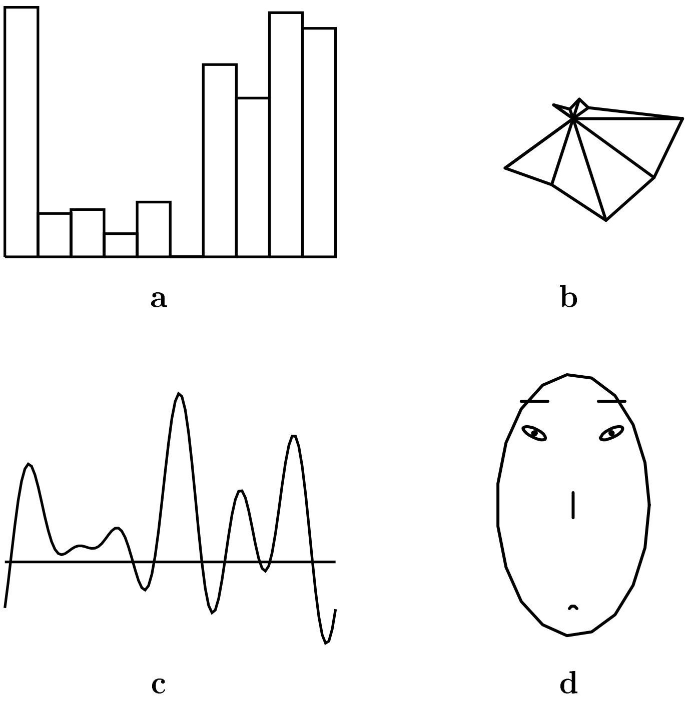

Figure 1: A ten-dimensional data item visualized using four different methods. **a** A profile of the component values, **b** a "star" in which the length of each ray emanating from the center illustrates one component, **c** Andrews' curve, and **d** a facial caricature.

kinds of summaries of the data set like the cluster centroids that are introduced below, or the reference vectors of a self-organizing map.

2.2 Clustering methods

The goal of clustering is to reduce the amount of data by categorizing or grouping similar data items together. Such grouping is pervasive in the way humans process information, and one of the motivations for using clustering algorithms is to provide automated tools to help in constructing categories or taxonomies (Jardine and Sibson, 1971; Sneath and Sokal, 1973). The methods may also be used to minimize the effects of human factors in the process.

Clustering methods (Anderberg, 1973; Hartigan, 1975; Jain and Dubes, 1988; Jardine and Sibson, 1971; Sneath and Sokal, 1973; Tryon and Bailey, 1973) can be divided into two basic types: hierarchical and partitional clustering. Within each of the types there exists a wealth of subtypes and different algorithms for finding the clusters.

Hierarchical clustering proceeds successively by either merging smaller clusters into larger ones, or by splitting larger clusters. The clustering methods differ in the rule by which it is decided which two small clusters are merged or which large cluster is split. The end result of the algorithm is a tree of clusters called a dendrogram, which shows how the clusters are related. By cutting the dendrogram at a desired level a clustering of the data items into disjoint groups is obtained.

Partitional clustering, on the other hand, attempts to directly decompose the data set into a set of disjoint clusters. The criterion function that the clustering algorithm tries to minimize may emphasize the local structure of the data, as by assigning clusters to peaks in the probability density function, or the global structure. Typically the global criteria involve minimizing some measure of dissimilarity in the samples within each cluster, while maximizing the dissimilarity of different clusters.

A commonly used partitional clustering method, K-means clustering (MacQueen, 1967), will be discussed in some detail since it is closely related to the SOM algorithm. In K-means clustering the criterion function is the average squared distance of the data items $\mathbf{x}_k$ from their nearest cluster centroids,

$$E_K = \sum_k \|\mathbf{x}_k - \mathbf{m}_{c(\mathbf{x}_k)}\|^2, \tag{1}$$

where $c(\mathbf{x}_k)$ is the index of the centroid that is closest to $\mathbf{x}_k$. One possible algorithm for minimizing the cost function begins by initializing a set of K cluster centroids denoted by $\mathbf{m}_i$, $i = 1, \ldots, K$. The positions of the $\mathbf{m}_i$ are then adjusted iteratively by first assigning the data samples to the nearest clusters and then recomputing the centroids. The iteration is stopped when E does not change markedly any more. In an alternative algorithm each randomly chosen sample is considered in succession, and the nearest centroid is updated.

Equation 1 is also used to describe the objective of a related method, vector quantization (Gersho, 1979; Gray, 1984; Makhoul et al., 1985). In vector quantization the goal is to minimize the average (squared) *quantization error*, the distance between a sample $\mathbf{x}$ and its representation $\mathbf{m}_{c(\mathbf{x})}$. The algorithm for minimizing Equation 1 that was described above is actually a straightforward generalization of the algorithm proposed by Lloyd (1957) for minimizing the average quantization error in a one-dimensional setting.

A problem with the clustering methods is that the interpretation of the clusters may be difficult. Most clustering algorithms prefer certain cluster shapes, and the algorithms will always assign the data to clusters of such shapes even if there were no clusters in the data. Therefore, if the goal is not just to compress the data set but also to make inferences about its cluster structure, it is essential to analyze whether the data set exhibits a clustering tendency. The results of the cluster analysis need to be validated, as well. Jain and Dubes (1988) present methods for both purposes.

Another potential problem is that the choice of the number of clusters may be critical: quite different kinds of clusters may emerge when K is changed. Good initialization of the cluster centroids may also be crucial; some clusters may even be left empty if their centroids lie initially far from the distribution of data.

Clustering can be used to reduce the amount of data and to induce a categorization. In exploratory data analysis, however, the categories have only limited value as such. The clusters should be illustrated somehow to aid in understanding what they are like. For example in the case of the K-means algorithm the centroids that represent the clusters are still high-dimensional, and some additional illustration methods are needed for visualizing them.

2.3 Projection methods

Clustering reduces the amount of data items by grouping them. There also exist methods that can be used for reducing the dimensionality of the data items, too. Such methods will be called *projection methods* below[2]. The goal of the projection is to represent the input data items in a lower-dimensional space in such a way that certain properties of the structure of the data set are preserved as faithfully as possible. The projection can be used to visualize the data set if a sufficiently small output dimensionality is chosen.

2.3.1 Linear projection methods

Principal component analysis (PCA) (Hotelling, 1933) can be used to display the data as a linear projection on such a subspace of the original data space that best preserves the variance in the data. It is a standard method in data analysis; it is well understood, and effective algorithms exist for computing the projection. Even neural algorithms exist (Oja, 1983; Oja, 1992; Rubner and Tavan, 1989; Cichocki and Unbehauen, 1993). A demonstration of PCA is presented in Figure 2.

Projection pursuit. In exploratory projection pursuit (Friedman, 1987; Friedman and Tukey, 1974) the data is projected linearly, but this time a projection which reveals as much of the non-normally distributed structure of the data set as possible is sought. This is done by assigning a numerical "interestingness" index to each possible projection, and by maximizing the index. The definition of interestingness is based on how much the projected data deviates from normally distributed data in the main body of its distribution. There is also a neural implementation of this idea (Fyfe and Baddeley, 1995).

After an interesting projection has been found, the structure that makes the projection interesting may be removed from the data, after which the procedure

[2]Ripley (1996) divides statistical data-analysis methods into clustering methods, projection methods, and multidimensional scaling (MDS) methods. Differentiation between the latter two is not useful here, since for patterns represented as vectors in a Euclidean space the MDS methods essentially form nonlinear projections.

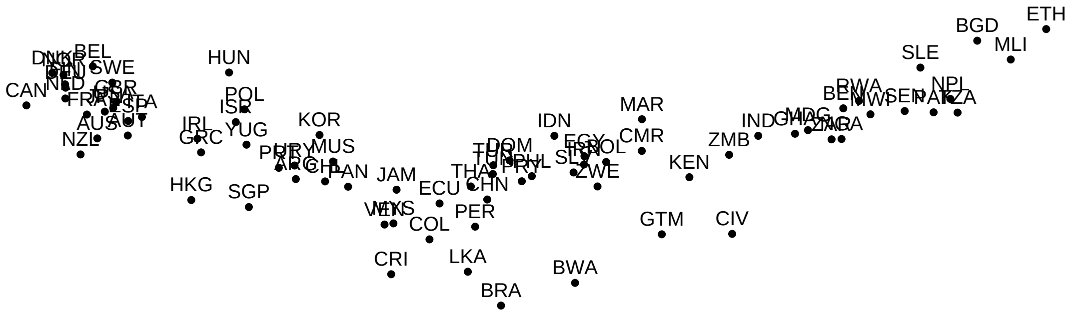

Figure 2: A dataset projected linearly onto the two-dimensional subspace obtained with PCA. Each 39-dimensional data item describes different aspects of the welfare and poverty of one country. The data set consisting of 77 countries, used also in Publication 2, was picked up from the World Development Report published by the World Bank (1992). Missing data values were neglected when computing the principal components, and zeroed when forming the projections. A key to the abbreviated country names is given in the Appendix.

can be restarted from the beginning to reveal more of the structure of the data set.

2.3.2 Nonlinear projection methods

PCA cannot take into account nonlinear structures, structures consisting of arbitrarily shaped clusters or curved manifolds, since it describes the data in terms of a linear subspace. Projection pursuit tries to express some nonlinearities, but if the data set is high-dimensional and highly nonlinear it may be difficult to visualize it with linear projections onto a low-dimensional display even if the "projection angle" is chosen carefully.

Several approaches have been proposed for reproducing nonlinear higher-dimensional structures on a lower-dimensional display. The most common methods allocate a representation for each data point in the lower-dimensional space and try to optimize these representations so that the distances between them would be as similar as possible to the original distances of the corresponding data items. The methods differ in how the different distances are weighted and how the representations are optimized.

Multidimensional scaling (MDS) refers to a group of methods that is widely used especially in behavioral, econometric, and social sciences to analyze subjective evaluations of pairwise similarities of entities, such as commercial products in a market survey. The starting point of MDS is a matrix consisting of the pairwise dissimilarities of the entities. In this thesis only distances between pattern vectors in a Euclidean space will be considered, but in MDS the dissimilarities need not be distances in the mathematically strict sense. In fact, MDS is perhaps most often used for creating a space where the entities can be represented as vectors, based on some evaluation of the dissimilarities of the entities.

The goal in this thesis is not merely to create a space which would represent the relations of the data faithfully, but also to reduce the dimensionality of the data set to a sufficiently small value to allow visual inspection of the set. The MDS methods can be used to fulfill this goal, as well.

There exists a multitude of variants of MDS with slightly different cost functions and optimization algorithms. The first MDS for metric data was developed in the 1930s (historical treatments and introductions to MDS have been provided by, for example, Kruskal and Wish, 1978; de Leeuw and Heiser, 1982; Wish and Carroll, 1982; Young, 1985), and later generalized for analyzing nonmetric data and even the common structure in several dissimilarity matrices corresponding to, for instance, evaluations made by different individuals.

The algorithms designed for analyzing a single dissimilarity matrix, and which can thus be used for reducing the dimensionality of a data set, can be broadly divided into two basic types, metric and nonmetric MDS.

In the original *metric MDS* (Torgerson, 1952; cf. Young and Householder, 1938) the distances between the data items have been given, and a configuration of points that would give rise to the distances is sought. Often a linear projection onto a subspace obtained with PCA is used. The key idea of the method, to approximate the original set of distances with distances corresponding to a configuration of points in a Euclidean space can, however, also be used for constructing a nonlinear projection method. If each item $\mathbf{x}_k$ is represented with a lower-dimensional, say, two-dimensional data vector $\mathbf{x}'_k$, then the goal of the projection is to optimize the representations so that the distances between the items in the two-dimensional space will be as close to the original distances as possible. If the distance between $\mathbf{x}_k$ and $\mathbf{x}_l$ is denoted by $d(k,l)$ and the distance between $\mathbf{x}'_k$ and $\mathbf{x}'_l$ in the two-dimensional space by $d'(k,l)$, the metric MDS tries to approximate $d(k,l)$ by $d'(k,l)$. If a square-error cost is used, the objective function to be minimized can be written as

$$E_M = \sum_{k \neq l} [d(k,l) - d'(k,l)]^2. \tag{2}$$

A perfect reproduction of the Euclidean distances may not always be the best possible goal, especially if the components of the data vectors are expressed on an ordinal scale. Then only the *rank order* of the distances between the vectors is meaningful, not the exact values. The projection should try to match the rank order of the distances in the two-dimensional output space to the rank order in the original space. The best possible rank ordering for a given configuration or points can be guaranteed by introducing a monotonically increasing function f that acts on the original distances, and always maps the distances to such values that best preserve the rank order. *Nonmetric MDS* (Kruskal, 1964; Shepard, 1962) uses such a function (Kruskal and Wish, 1978), whereby the normalized cost function becomes

$$E_N = \frac{1}{\sum_{k \neq l} [d'(k,l)]^2} \sum_{k \neq l} [f(d(k,l)) - d'(k,l)]^2. \tag{3}$$

For any given configuration of the projected points $\mathbf{x}'_k$, f is always chosen to minimize Equation 3.

Although the nonmetric MDS was motivated by the need of treating ordinal-scale data, it can also be used of course if the inputs are presented as pattern vectors in a Euclidean space. The projection then only tries to preserve the order of the distances between the data vectors, not their absolute values. A demonstration of nonmetric MDS, applied in a dimension reduction task, is given in Figure 3.

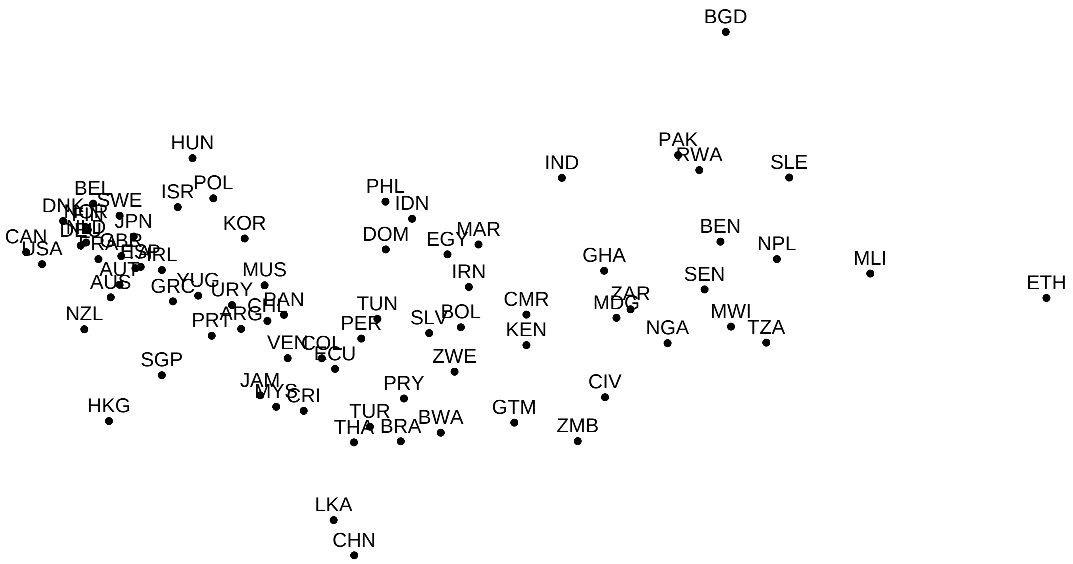

Figure 3: A nonlinear projection constructed using nonmetric MDS. The data set is the same as in Figure 2. Missing data values were treated by the following simple method, which has been demonstrated to produce good results at least in the pattern recognition context (Dixon, 1979). When computing the distance between a pair of data items, only the (squared) differences between component values that are available are computed. The rest of the differences are then set to the average of the computed differences.

Another nonlinear projection method, *Sammon's mapping* (Sammon, Jr., 1969), is closely related to the metric MDS version described above. It, too, tries to optimize a cost function that describes how well the pairwise distances in a data set are preserved. The cost function of Sammon's mapping is (omitting a constant normalizing factor)

$$E_S = \sum_{k \neq l} \frac{[d(k,l) - d'(k,l)]^2}{d(k,l)}. \tag{4}$$

The only difference between Sammon's mapping and the (nonlinear) metric MDS (Eq. 2) is that the errors in distance preservation are normalized with the distance in the original space. Because of the normalization the preservation of small distances will be emphasized.

A demonstration of Sammon's mapping is presented in Figure 4.

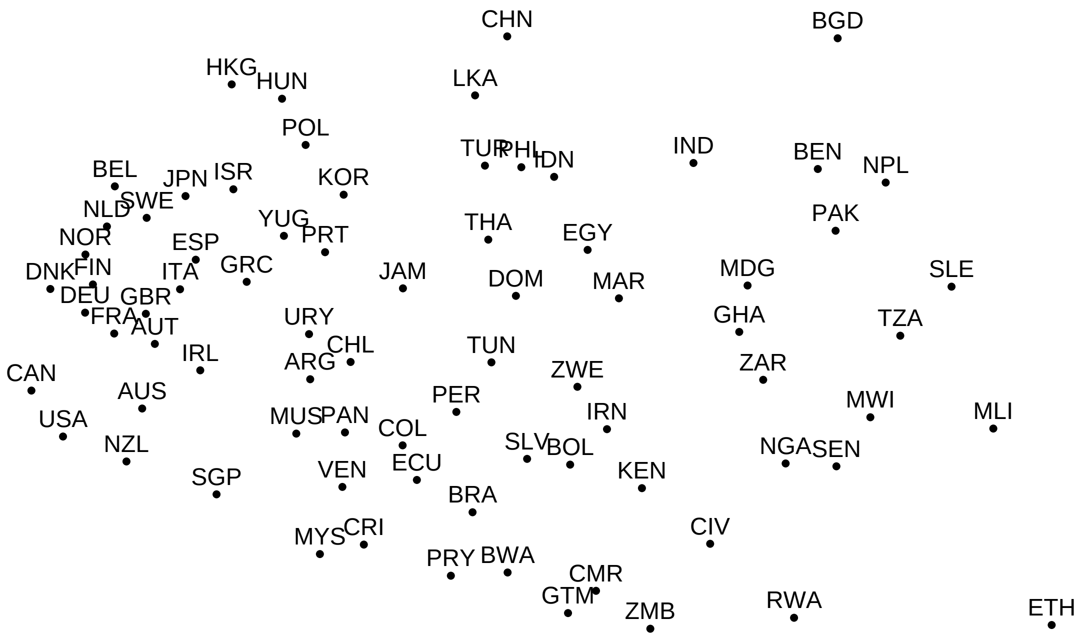

Figure 4: Sammon's mapping of the data set which has been projected using PCA in Figure 2 and nonmetric MDS in Figure 3. Missing data values were treated in the same manner as in forming the nonmetric MDS.

Principal curves. PCA can be generalized to form nonlinear curves. While in PCA a good projection of a data set onto a linear manifold was constructed, the goal in constructing a principal curve is to project the set onto a nonlinear manifold. The principal curves (Hastie and Stuetzle, 1989) are smooth curves that are defined by the property that each point of the curve is the average of all data points that project to it, i.e., for which that point is the closest point on the curve. Intuitively speaking, the curves pass through the "center" of the data set. Principal curves are generalizations of principal components extracted using PCA in the sense that a linear principal curve is a principal component; the connections between the two methods are delineated more carefully in the original article. Although the extracted structures are called principal *curves* the generalization to surfaces seems relatively straightforward, although the resulting algorithms will become computationally more intensive.

The conception of continuous principal curves may aid in understanding how principal components could be sensibly generalized. To be useful in practical computations, however, the curves must be discretized. It has turned out (Mulier and Cherkassky, 1995; Ritter et al., 1992) that discretized principal curves are essentially[3] equivalent to SOMs, introduced before Hastie and Stuetzle (1989) introduced the principal curves. It thus seems that the conception of principal curves is most useful in providing one possible viewpoint to the properties of the SOM algorithm.

Other methods. A problem with the nonlinear MDS methods is that they are computationally very intensive for large data sets. The computational complexity can be reduced, however, by restricting attention to a subset of the distances between the data items. When placing a point on a plane its distance from two other points of the plane can be set exactly. This property is used in the *triangulation method* (Lee et al., 1977). Points are mapped sequentially onto the plane, and the distance of the new item to the two nearest items already mapped is preserved. Alternatively, the distance to the nearest item and a reference point that is common to all items may be preserved. The points are mapped in such an order that all of the nearest-neighbor distances in the original space will be preserved. The triangulation can be computed quickly, compared to the MDS methods, but since it only tries to preserve a small fraction of the distances the projection may be difficult to interpret for large data sets. The method may, however, be useful in connection with Sammon's mapping (Biswas et al., 1981).

The dimensionality of data sets can also be reduced with the aid of autoassociative neural networks that represent their inputs using a smaller number of variables than there are dimensions in the input data. Such networks try to reconstruct their inputs as faithfully as possible, and the representation of the data items constructed into the network can be used as the reduced-dimensional expression of the data. Some linear and nonlinear associative memories have been introduced by Kohonen (1984). The representations formed into the hidden layer

[3]The algorithm Hastie and Stuetzle (1989) proposed for finding discretized principal curves resembles the batch version (Kohonen, 1995c) of the SOM algorithm, although the details are different.

of a multilayer perceptron have also been used for the dimension reduction task (DeMers and Cottrell, 1993; Garrido et al., 1995). A special version of the multilayer perceptrons, a *replicator neural network* (Hecht-Nielsen, 1995) has even been shown *capable* of representing its inputs in terms of their "natural coordinates". This occurs for a somewhat idealized model when the inherent dimensionality q of the data increases. The natural coordinates correspond to coordinates in a q-dimensional unit cube that has been transformed elastically to fit the distribution of the data. The inherent dimensionality of the data is, of course, difficult to identify in practice.

The replicator neural networks could possibly be used for forming a visualization of the data set by choosing $q = 2$. Although intriguing, the approach would require a separate study that would compare both the quality of the results and the computational requirements for a network having a practical size. The learning of the multilayer perceptrons with the backpropagation algorithm (cf., e.g., Rumelhart et al., 1986) is known to be very slow (Haykin, 1994), but it is possible that some alternative learning algorithms would be more feasible.

2.4 Self-organizing maps

The self-organizing map (SOM) (Kohonen, 1982; Kohonen, 1990; Kohonen, 1995c; Kohonen et al., 1996b) is a neural network algorithm that has been used for a wide variety of applications, mostly for engineering problems but also for data analysis (e.g., Back et al., 1996; Blayo and Demartines, 1992; Carlson, 1991; Cheng et al., 1994; Garavaglia, 1993; Martín-del-Brío and Serrano-Cinca, 1993; Marttinen, 1993; Serrano-Cinca, 1996; Ultsch, 1993b; Ultsch and Siemon, 1990; Varfis and Versino, 1992; Zhang and Li, 1993). A comprehensive treatment of the topic is provided by Kohonen (1995c); here only aspects relevant for data exploration and aspects needed for understanding the relationships between the SOM and the other algorithms will be presented.

In this section the data mining tools have been divided into two categories, clustering methods and projection methods. The SOM is a special case in that it can be used at the same time both to reduce the amount of data by clustering, and for projecting the data nonlinearly onto a lower-dimensional display.

The basic algorithm is first motivated with a discussion of competitive learning in Section 2.4.1. Some of its properties that are useful for data analysis are then introduced in Section 2.4.2. Mathematical treatments of the algorithm, and its relations to other algorithms are discussed in Section 2.4.3. Especially two algorithms, the K-means clustering and the principal curves, are very closely related to the SOM. A mathematically oriented reader may wish to concentrate on Section 2.4.3; the algorithm is fully usable without any reference to competitive learning.

2.4.1 The self-organizing map algorithm

Competitive learning is an adaptive process in which the neurons in a neural network gradually become sensitive to different input categories, sets of samples in a specific domain of the input space (Amari, 1980; Didday, 1970; Didday, 1976; Grossberg, 1976; Kohonen, 1982; Kohonen, 1984; Nass and Cooper, 1975; Pérez et al., 1975; Swindale, 1980; von der Malsburg, 1973). A kind of a division of labor emerges in the network when different neurons specialize to represent different types of inputs.

The specialization is enforced by competition among the neurons: when an input $\mathbf{x}$ arrives, the neuron that is best able to represent it wins the competition and is allowed to learn it even better, as will be described below.

If there exists an ordering between the neurons, i.e., the neurons are located on a discrete lattice, the self-organizing map, the competitive learning algorithm can be generalized: if not only the winning neuron but also its *neighbors* on the lattice are allowed to learn, neighboring neurons will gradually specialize to represent similar inputs, and the representations will become *ordered* on the map lattice. This is the essence of the SOM algorithm.

The neurons represent the inputs with reference vectors $\mathbf{m}_i$, the components of which correspond to synaptic weights. One reference vector is associated with each neuron called *unit* in a more abstract setting. The unit, indexed with c, whose reference vector is nearest to the input $\mathbf{x}$ is the winner of the competition:

$$c = c(\mathbf{x}) = \arg\min_i \{\|\mathbf{x} - \mathbf{m}_i\|^2\} \ . \tag{5}$$

Usually Euclidean metric is used, although other choices are possible as well.

The winning unit and its neighbors adapt to represent the input even better by modifying their reference vectors towards the current input. The amount the units learn will be governed by a neighborhood kernel h, which is a decreasing function of the distance of the units from the winning unit on the map lattice. If the locations of units i and j on the map grid are denoted by the two-dimensional vectors $\mathbf{r}_i$ and $\mathbf{r}_j$, respectively, then $h_{ij}(t) = h(\|\mathbf{r}_i - \mathbf{r}_j\|; t)$, where t denotes time.

During the learning process at time t the reference vectors are changed iteratively according to the following adaptation rule, where $\mathbf{x}(t)$ is the input at time t and $c = c(\mathbf{x}(t))$ is the index of the winning unit:

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + h_{ci}(t)[\mathbf{x}(t) - \mathbf{m}_i(t)] \ . \tag{6}$$

In practice the neighborhood kernel is chosen to be wide in the beginning of the learning process to guarantee global ordering of the map, and both its width and height decrease slowly during learning.

The learning process consisting of winner selection by Equation 5 and adaptation of the synaptic weights by Equation 6, can be modeled with a neural network structure, in which the neurons are coupled by inhibitory connections (Kaski and

Kohonen, 1994; Kohonen, 1993).

2.4.2 Properties useful in exploring data

By virtue of its learning algorithm the SOM forms a nonlinear regression of the ordered set of reference vectors into the input space. The reference vectors form a two-dimensional "elastic network" that follows the distribution of the data.

Ordered display. The ordered nature of the regression justifies the use of the map as a display for data items. When the items are mapped to those units on the map that have the closest reference vectors, nearby units will have similar data items mapped onto them. Such an *ordered display* of the data items facilitates understanding of the structures in the data set. Kohonen (1981) was the first to propose using such displays to illustrate a data set.

The same display can be used for displaying several other kinds of information. One clear advantage of always using the same display is that as the analysts grow more familiar with the map, they can interpret new information displayed on it faster and more easily.

For example, the map display can be used as an *ordered groundwork* on which the original data variables, components of the data vectors, can be displayed in their natural order. Such displays have been demonstrated in Publication 2. The variables become smoothed locally on the display, which helps in gaining insight in the distributions of their values in the data set. Such displays are much more illustrative than, for instance, raw linearly organized statistical tables. It might also be useful to display the *residuals*, average differences of the variables from their smoothed values.

Visualization of clusters. The same ordered display can be used for illustrating the *clustering density* in different regions of the data space. The density of the reference vectors of an organized map will reflect the density of the input samples (Kohonen, 1995c; Ritter, 1991). In clustered areas the reference vectors will be close to each other, and in the empty space between the clusters they will be more sparse. Thus, the cluster structure in the data set can be brought visible by displaying the distances between reference vectors of neighboring units (Kraaijveld et al., 1992; Kraaijveld et al., 1995; Ultsch, 1993b; Ultsch and Siemon, 1990).

The cluster display may be constructed as follows (Iivarinen et al., 1994). The distance between each pair of reference vectors is computed and scaled so that the distances fit between a given minimum and maximum value, after optionally removing outliers. On the map display each scaled distance value determines the gray level or color of the point that is in the middle of the corresponding map units. The gray level values in the points corresponding to the map units themselves are set to the average of some of the nearest distance values (on a hexagonal grid, e.g., to the average of three of the six distances toward the lower-right corner). After these values have been set up, they can be visualized as such on the display, or smoothed spatially.

The resulting cluster diagram is very general in the sense that nothing needs to be assumed about the shapes of the clusters. Most of the clustering algorithms prefer clusters of certain shapes (Jain and Dubes, 1988).

A demonstration of a display constructed using SOM is presented in Figure 5.

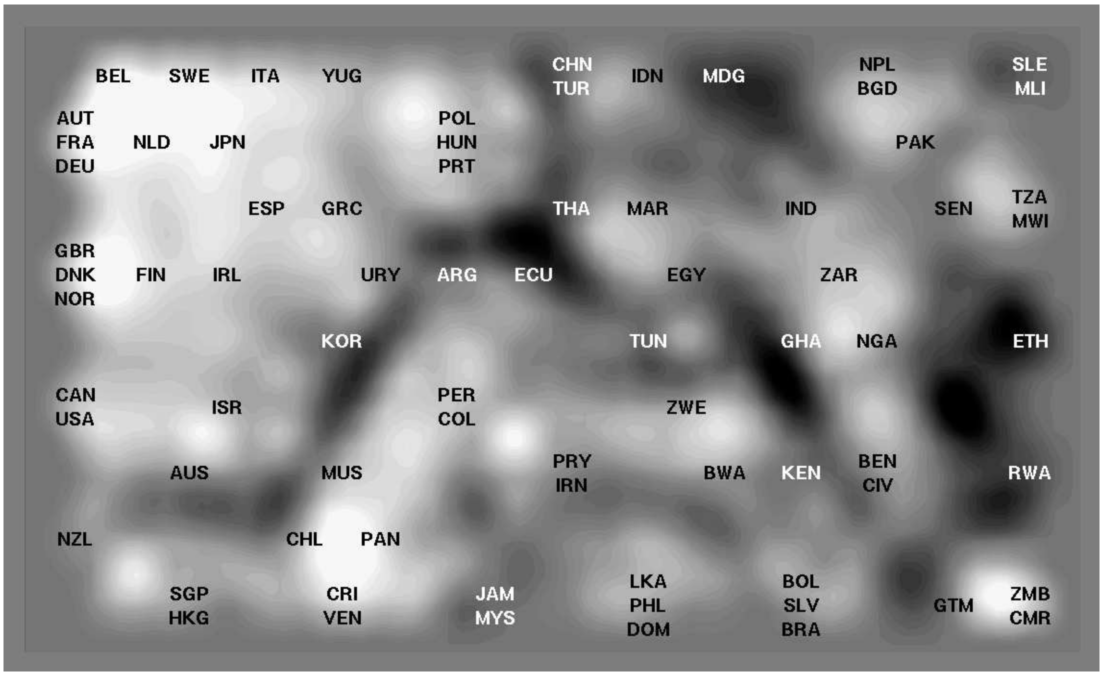

Figure 5: A map display constructed using the SOM algorithm. The overall order of the countries seems to correspond fairly closely to the Sammon's mapping of the same data set (Fig. 4). The most prominent clustering structures are also visible in both displays. Details on how the map was constructed are presented in Publication 2. The size of the map was 13 by 9 units.

Missing data. A frequently occurring problem in applying methods of statistics is that of missing data. Some of the components of the data vectors are not available for all data items, or may not even be applicable or defined. Several simple (e.g., Dixon, 1979) and more complex (e.g., Dempster et al., 1977) approaches have been proposed for tackling this problem, from which all of the clustering and projection methods suffer likewise.

In the case of the SOM the problem of missing data can be treated as follows: when choosing the winning unit by Equation 5, the input vector $\mathbf{x}$ can be compared with the reference vectors $\mathbf{m}_i$ using only those components that are available in $\mathbf{x}$. Note that none of the reference vector components is missing. If only a small proportion of the components of the data vector is missing, the result of the comparison will be statistically fairly accurate. When the reference vectors are then adapted using Equation 6, only the components that are available in $\mathbf{x}$ will be modified.

It has been demonstrated that better results can be obtained with the approach described above than by discarding the data items from which components are missing (Samad and Harp, 1992). However, for data items from which the majority of the indicators are missing it is not justifiable to assume that the winner selection is accurate. A reasonable compromise, used in Publication 2, is to discard data items with too many (exceeding a chosen proportion) missing values from the learning process. Even the discarded samples can, however, be tentatively displayed on the map after it has been organized.

Note: Although the SOM as such can be used to explore incomplete data sets, some preprocessing methods may have problems with missing components of the input data items. For example, normalization of the data *vectors* cannot be done in a straightforward manner. Normalization of the variance of each component separately is, in contrast, a viable operation even for incomplete data sets.

Outliers. In measurement data there may exist *outliers*, data items lying very far from the main body of the data. The outliers may result, for instance, from measurement errors or typing errors made while inserting the statistics into a data base. In such cases it would be desirable that the outliers would not affect the result of the analysis. This is indeed the case for map displays generated by the SOM algorithm: each outlier affects only one map unit and its neighborhood, while the rest of the display may still be used for inspecting the rest of the data. Furthermore, the outliers can be easily detected based on the clustering display: the input space is, by definition, very sparsely populated near the outliers. If desired, the outliers can then be discarded and the analysis can be continued with the rest of the data set.

It is also possible that the outliers are not erroneous but that some data items really are strikingly different from the rest. In any case the map display reveals the outliers, whereby they can either be discarded or paid special attention to.

2.4.3 Mathematical characterizations

Rigorous mathematical treatment of the SOM algorithm has turned out to be extremely difficult in general (reviews have been provided by Kangas, 1994; and Kohonen, 1995c). In the case of a discrete data set and a fixed neighborhood kernel, however, there exists a potential function for the SOM, namely (Kohonen, 1991; Ritter and Schulten, 1988)

$$E = \sum_k \sum_i h_{ci} \|\mathbf{x}_k - \mathbf{m}_i\|^2 \,, \tag{7}$$

where the index c depends on the $\mathbf{x}_k$ and the reference vectors $\mathbf{m}_i$ (cf. Eq. 5).

The learning rule of the SOM, Equation 6, corresponds to a gradient descent step in minimizing the sample function

$$E_1 = \sum_i h_{ci} \|\mathbf{x}(t) - \mathbf{m}_i\|^2 \tag{8}$$

obtained by selecting randomly a sample $\mathbf{x}(t)$ at iteration t. The learning rule then corresponds to a step in the stochastic approximation of the minimum of Equation 7, as discussed by Kohonen (1995c).

Note: In Equation 7 the index c is a function of all the reference vectors, which implies that it may change when the gradient descent step is taken. Locally, if the index $c = c(\mathbf{x}_k)$ does not change for any $\mathbf{x}_k$, the gradient step is valid, however.

Relation to K-means clustering. The cost function of the SOM, Equation 7, closely resembles Equation 1, which the K-means clustering algorithm tries to minimize. The difference is that in the SOM the distance of each input from all of the reference vectors instead of just the closest one is taken into account, weighted by the neighborhood kernel h. Thus, the SOM functions as a conventional clustering algorithm if the width of the neighborhood kernel is zero.

The close relation between the SOM and the K-means clustering algorithm also hints at why the self-organized map follows rather closely the distribution of the data set in the input space: it is known for vector quantization that the density of the reference vectors approximates the density of the input vectors for high-dimensional data (Kohonen, 1995c; Zador, 1982), and K-means is essentially equivalent to vector quantization. In fact, an expression for the density of the reference vectors of the SOM has been derived in the one-dimensional case (Ritter, 1991); in the limit of a very wide neighborhood and a large number of reference vectors the density is proportional to $p(\mathbf{x})^{2/3}$, where $p(\mathbf{x})$ is the probability density function of the input data.

Note: Although the K-means clustering algorithm and the SOM are very closely related, the best ways of using them in data mining are probably different. Whereas in the K-means clustering algorithm the number K of clusters should be chosen according to the number of clusters there are in the data, in the SOM the number of reference vectors can be chosen to be much larger, irrespective of the number of clusters. The cluster structures will become visible on the special displays that were discussed in Section 2.4.2.

Relation to principal curves. The SOM algorithm creates a representation of the input data set that follows the data distribution. The representation of the data set is also organized. One possible view (cf. Ritter et al., 1992) to the organization has been provided by the mathematical characterization of principal curves (Hastie and Stuetzle, 1989).

Each point on a principal curve is the average of all the points that project to it. The curve is thus formed of conditional expectations of the data. For discrete distributions it is not sensible to define such curves, but it is possible to construct practical algorithms if the data is "smeared" spatially. In the SOM a similar smearing is performed by the neighborhood kernel in the adaptation process (cf. Eq. 6). It is well-known in SOM literature that in the SOM each reference vector represents local conditional expectations of the data items — the batch map algorithm (Kohonen, 1995c) is essentially a manifestation of this idea. The principal curves or manifolds are thus essentially continuous counterparts of the SOM.

The principal curves also have another characterization which, based on the previous discussion, may be used as one source in providing an intuitive understanding of the SOM algorithm. The goodness of a curve in representing a data distribution can be measured, for example, by the average (squared) distance of the data points from the curve, like the goodness of the K-means algorithm was measured by the average (squared) distance of the data points from the nearest cluster. The principal curves are the *critical points* of this measure, i.e., they are extremal with respect to small, smooth variations (Hastie and Stuetzle, 1989). This implies that any smooth curve which corresponds to a minimum of the average distance from the data items is a principal curve.

A decomposition of the cost function. The cost function of the SOM, Equation 7, can be decomposed into two terms as follows (Lampinen and Oja, 1992; here a discrete version will be presented):

$$E = \sum_k \|\mathbf{x}_k - \mathbf{n}_c\|^2 + \sum_i \sum_j h_{ij} N_i \|\mathbf{n}_i - \mathbf{m}_j\|^2 \ . \tag{9}$$

Here N_i denotes the number of the data items which are closest to the reference vector $\mathbf{m}_i$, and $\mathbf{n}_i = 1/N_i \sum_{\mathbf{x}_k \in V_i} \mathbf{x}_k$, where V_k is the Voronoi region corresponding to the reference vector $\mathbf{m}_i$. When deriving this approximation is was assumed that $\sum_j h_{ij} = 1$ for all i, which holds exactly for toroidal maps when the kernel h has the same shape for all i, and also away from the borders on a non-toroidal map if the kernel differs from zero only locally.

The first term in Equation 9 corresponds to the cost function of the K-means clustering algorithm, that is, the average distance from the data points to the nearest cluster centroid. Here the clusters are not defined in terms of their centroids, however, but in terms of the reference vectors $\mathbf{m}_i$. This first term may be interpreted as one way of measuring how accurately the map follows the distribution of the input data.

The second term, on the other hand, may be interpreted as governing the ordering of the reference vectors. In considering the second term it may be of help to note that $\mathbf{n}_i$ and $\mathbf{m}_i$ will in general be close to each other, since $\mathbf{n}_i$ is the centroid of the cluster *defined* by $\mathbf{m}_i$. At a fixed point of the SOM algorithm $\mathbf{m}_i = \sum_k h_{c(\mathbf{x}_k)i}\mathbf{x}_k / \sum_k h_{c(\mathbf{x}_k)i}$, which is closer to $\mathbf{n}_i$ the more the neighborhood kernel h_{ci} is centered around c. To minimize the second term the units that are close to each other on the map should have similar reference vectors, since the value of the neighborhood kernel is large. Units that lie farther away on the map may, on the other hand, have quite dissimilar reference vectors, since the neighborhood kernel is small and the distance thus does not make a large contribution to the error.

2.4.4 Some variants

A rich variety of versions of the basic SOM algorithm have been proposed (cf. Kohonen, 1995c). Some of the variants aim at improving the preservation of topology by using more flexible map structures instead of the fixed grid (Blackmore and Miikkulainen, 1993; Bruske and Sommer, 1995; Fritzke, 1991; Fritzke, 1994; Martinetz and Schulten, 1991; Martinetz and Schulten, 1994). While some of these methods may be useful for other purposes, they cannot be used for visualization, at least not as easily as the regular grid.

Some other variants aim at reducing the computational complexity of the SOM. The speed of computation is an extremely important aspect in data mining when vast databases are analyzed. In Publication 4 we used computational speedups that have been developed by Kohonen; in the other studies the basic SOM algorithm was used. Other speedup methods have also been proposed. For example, a one-dimensional map can be enlarged simply by inserting a reference vector between each pair of neighbors (Luttrell, 1988).

The search for the best-matching unit can be speeded up by constructing a tree-structured SOM (Koikkalainen, 1994; Koikkalainen, 1995; Koikkalainen and Oja, 1990), where each level of the tree consists of a separate, progressively larger SOM. The search for the best match then proceeds level by level, at each time restricting the search to a subset of units that is governed by the location of the best match in the previous, smaller level. The map is taught one level at a time, starting from the smallest level. During teaching the best match search can be done even more quickly if the data set is relatively small: the location of the best match in the previous level can be tabulated for each input sample. The subset of units in which the search for a winner needs to be performed can then be found by a single table lookup.

Yet another fast approach is to construct a hierarchical tree of SOMs, where the SOMs in the bottom layer treat different subsets of the components of the input variables. The outputs of the SOMs in the bottom layer are then combined in a hierarchical manner towards the final SOM that takes into account the whole input vector (Luttrell, 1989). Since each of the small SOMs receives only very small-dimensional input vectors, the winning unit can potentially be sought with a single, fast table lookup.

Each of the computational speedups could potentially be useful, but the comparison of the methods would require a careful study. Most of the truly effective speedups are suboptimal in the sense that they only approximate the original SOM, and a careful study is therefore needed to guarantee that the results are satisfactory for a given application.

Bishop et al. (1996a, 1996b) have recently introduced a latent-variable density model called Generative Topographic Mapping (GTM), which is closely related to the SOM and to principal curves. In latent variable models it is assumed that the data set can be explained using a small set of latent variables. In principle the latent variable space could be continuous, but in GTM a discrete grid like the grid of the SOM is used for computational reasons. A probability distribution on the latent grid is postulated to generate a probability distribution in the input space through a parameterized model. Given an input data set, the goal of the GTM is then to fit the model, in the maximum likelihood sense, to the data set. This is done using the expectation-maximization (EM) algorithm (Dempster et al., 1977), by iteratively estimating first the probability distribution on the latent grid and then maximizing the likelihood of the input data, given the distribution.

The GTM is claimed to have several advantageous properties when compared with the SOM, and no significant disadvantages (Bishop et al., 1996b). It may be useful to study these properties more closely to get a deeper understanding of the relations and relative merits of the methods.

The GTM has an objective function, which may facilitate theoretical analyses of the method. The SOM does not have an objective function if the input data distribution is continuous (Erwin et al., 1992). In practical applications the data sets are always finite, however, and the SOM does have a (local) objective function, Equation 7.

The methods differ in that the GTM does not have a neighborhood function which governs the smoothness of the mapping in the SOM algorithm. In GTM the smoothness is governed by the widths of certain basis functions that are used in generating the probability distribution in the input space. The only essential difference regarding the neighborhood then seems to be that in the SOM the width of the neighborhood kernel is decreasing in time according to certain, well-established schedules (cf., e.g., Kohonen, 1995c; Kohonen et al., 1996a) whereas in GTM the basis functions remain fixed.

One possible way of defining topology preservation is to require that the mapping from the lower-dimensional output space to the input space is smooth and continuous. According to this definition at least a continuous version of GTM can be interpreted to be topology preserving. Since even smooth, continuous mappings can be very "twisted", however, a measure of the regularity of the mapping is also needed. Most of the attempts to measure the topology preservation of the SOMs have in fact aimed at measuring the goodness (or regularity) of the mapping; some approaches will be discussed in Section 3.3. Measures that help in choosing basis functions that provide suitably regular mappings would probably be useful in the case of the GTM, as well.

It is probably useful in practical applications that different mappings generated with the GTM algorithm can be compared by comparing their likelihoods. Likelihoods have a straightforward interpretation. Neither the objective function of the SOM, Equation 7, nor the goodness measure presented in Publication 7 can be interpreted in this manner. These methods enable, however, selecting the best mapping from among a set of candidate mappings, which is the main goal in most applications.

It remains to be seen how important the advantages of the GTM are in practical applications. In addition to the advantages there is one undisputed disadvantage, however: the computational complexity. Computation of the GTM requires almost twice the time of the SOM (Bishop et al., 1996b). Moreover, this difference is probably enhanced if the speedup methods discussed in this section are introduced, since it may be difficult to apply similar speedups to GTM. Most of the speedup methods reduce the computational complexity of the best match search; there are, however, no best matching units in GTM since all units contribute directly in representing each input.

2.4.5 Notes on statistical accuracy

The question of how large the SOM grid should be has a simple answer if there is an unlimited number of learning samples available: as large as the available computational resources allow[4]. In Publication 1 this is essentially the case.

If the data set consists of a finite sample from a larger data set, then the question of the statistical accuracy of the map must be considered, i.e., whether the correct positions of the reference vectors can be estimated accurately based on the available data.

The question is especially important if the map is to be used later for displaying new data items. If the new items can be assumed to follow the same distribution as the items that were present while learning, and if the map is statistically accurate, then the new items can be displayed as accurately as the old ones. In high-dimensional spaces it is in general very difficult to achieve sufficient statistical accuracy since the samples are necessarily sparse. By fitting lower-dimensional structures like the SOM grid to the data set in an unsupervised manner, however, there is some hope of a generalizability of the results.

A possible rule-of-thumb for choosing a suitable size for the map might be that the number of free parameters should be at most, say, a certain fraction of the number of values used for estimating them. It seems, however, that it is difficult to compute the number of free parameters in the SOM, since the neighborhood kernel restricts the placement of the reference vectors. A rule-of-thumb that is

[4]Increase in the map size brings more resolution into the mapping. The "stiffness" of the map, or smoothness of the mapping, can be controlled independently of the map size by changing the final width of the learning neighborhood.

based directly on the total number of parameters in the SOM may then be overly conservative.

If the goal of data analysis is merely to visualize a given data set for exploratory purposes as in Publication 2, and if no new samples need to be mapped later, then the map can be taught to represent the given input data set with a sufficient statistical accuracy by sampling with replacement even from a small data set during the computation of the map. Then, of course, there is no guarantee that the map will generalize well to new data. The SOM is useful, however, even for maps having more units than there are data items. Even if the map passed through all data items, it would do so in an ordered manner and the distances between close-by data item pairs could be visualized on the map displays using the methods discussed in Section 2.4.2.

A brief introduction to one application of the SOM may serve to illustrate this point. A one-dimensional circular SOM can in effect be used for finding an approximate solution to the traveling salesman problem (Angéniol et al., 1988; Budinich, 1996): find the shortest route that passes through all of the data items once. The map forms a path that follows the data distribution, i.e., extends close to each data item. A two-dimensional map follows the data in a similar manner. It forms an "elastic network" that can follow the data the more closely the more there are map units. (Note, however, that the "smoothness" or regularity of the mapping can be controlled independently of the map size by changing the width of the neighborhood kernel.)

Setting the number of nodes approximately equal to the number of the input samples seems to be a useful rule-of-thumb for many applications when the data sets are relatively small. More extensive empirical investigations should be done, however, for example by cross-validating the results utilizing either the cost function (Eq. 7) or the measures presented in Publication 7.

2.5 Relations and differences between SOM and MDS

Relations of the mappings in the ideal case. The relative performance of the different algorithms in reducing the dimensionality of the input data has been studied in several articles (Bezdek and Pal, 1995; Flexer, 1997; Goodhill et al., 1995; Kraaijveld et al., 1992; Kraaijveld et al., 1995; Mao and Jain, 1995). In these studies the measures of performance were, however, mostly related to the preservation of the distances between points. In none of the studies has a cost function of the SOM-type (Eq. 7) been considered although for example the cost function of the Sammon's mapping (Eq. 4) has been used. It may not be surprising that Sammon's mapping is a good method according to that measure. Goodhill et al. (1995), on the other hand, define the preservation of neighborhoods as the preservation of distance ordering, which is the goal of nonmetric MDS.

The approach of comparing pattern classification capability in the original and the mapped space (Kraaijveld et al., 1992; Kraaijveld et al., 1995) might be a valid measure of feature extraction performance, but unfortunately it cannot be used for unlabeled data sets.

Since the methods pursue different although related goals, a more productive approach than trying to quantify their relative performance may be to analyze their differences qualitatively. Below, a short nonformal delineation of the differences between the SOM and the other methods is presented, based on the differences of their cost functions.

The relation between the SOM and clustering methods on one hand, and the nonlinear distance-preserving projection methods on the other, is perhaps best revealed by the decomposition of the SOM cost function, Equation 9. The SOM both performs clustering (first term in Eq. 9) and organizes the clusters (second term in Eq. 9); it can thus be used, at the same time, both for reducing the amount of data and for projecting it onto a low-dimensional display.

The essential difference between the projection formed by the SOM and the nonlinear distance-preserving projection methods (MDS) seems to be that the SOM tries to form a locally correct projection, whereas the MDS methods try to directly preserve all interpoint distances. This applies also to nonmetric MDS; although it only tries to preserve the rank order of the distances, it nevertheless treats all distances equally.

Sammon's mapping emphasizes the preservation of the local distances. The SOM does not, however, try to *preserve* the distances, but instead forms a kind of a homeomorphism, an "order-preserving nonlinear regression" of the map grid into the data set. The order is determined by the neighborhood kernel. If the kernel is localized (narrow), the order is determined *locally*. Global order arises because of the local interactions and because the gradual narrowing of the neighborhood kernel aids in avoiding "unordered" configurations.

In the general case a high-dimensional space cannot of course be accurately mapped onto a low-dimensional one by any method. In such cases the results of the SOM and those of the distance-preserving projection methods may be quite different. Since it is not possible to preserve all distances, it is quite possible that the MDS methods preserve most distances approximately and some distances poorly. Especially in the metric MDS the long distances will dominate over the shorter, local ones. In the SOM, in contrast, the localized neighborhood kernel determines that only the local distances will contribute to the error function.

In terms of exploratory data analysis the difference might be interpreted as follows: in the MDS methods the coarse global order of the projected data items will probably be more accurate and they, by definition, provide a more accurate view of the distances between the data items. The SOM, in contrast, tries to guarantee that items projected to nearby locations are similar, whereby the local order and local clustering structures shown on the map display are always as trustworthy as possible. Global structures will usually be displayed as well, but the local ones are considered to be more important.

The difference therefore seems to be that MDS tries to preserve the *metric* (or, in the case of nonmetric MDS, the global ordering relations) of the original space, whereas the SOM tries to preserve the *topology*, i.e., the local neighborhood relations. Sammon's mapping lies in the middle: it tries to preserve the metric, but considers preservation of local structures more important than the other MDS methods do.

It may be hard to define topology preservation rigorously for finite discrete structures, but it may still be possible to construct viable practical measures of how well the neighborhood relations are preserved. Such measures are discussed in more detail in Section 3.3 and in Publication 7.

The difference between methods that preserve distances and methods that preserve topology can perhaps be clarified through a hypothetical experiment with a data set consisting of a curved two-dimensional surface in a three-dimensional space. Distance-preserving methods require three dimensions to describe the structure of the data adequately. Topology-preserving methods like the SOM, on the other hand, only need two dimensions. The "elastic network" that the map forms can follow the curved surface in the data space.

Computational properties. It is common to all MDS methods that they *do not construct an explicit mapping function*, but instead the projections of all samples are computed in a simultaneous optimization process. Thus, new samples cannot be projected without recomputing the whole projection, at least not accurately.

At least in the case of Sammon's mapping this problem may be alleviated somewhat by computing the projection with a multilayer perceptron-type network which learns to minimize the cost function. This can be done by using a learning rule that is analogous to the error backpropagation algorithm (Mao and Jain, 1995). The method does not necessarily reduce the computational complexity of the original algorithm since the backpropagation learning rule is computationally very intensive, but it generalizes the mapping to new samples. An alternative approach is to use a radial basis function (RBF) network to construct the distance-preserving mapping (Webb, 1995); this could be a promising alternative to the original MDS methods.

The optimization of the cost functions of the MDS methods requires comparisons between all pairs of vectors, whereby their computational complexity is of the order of N^2, $\mathcal{O}(N^2)$, where N is the number of data items. The computational complexity of the SOM is K^2, where K is the number of map units: each learning step requires $\mathcal{O}(K)$ computations, and to achieve a sufficient statistical accuracy the number of iterations should be at least some multiple of K.

If the size of the map is of the order of the number of input samples as in Publication 2, the computational complexities of the methods are of the same order of magnitude. If reduction of complexity is needed, the resolution of the SOM can be reduced by using a smaller map. It is also possible to use faster versions of the algorithm, discussed in Section 2.4.4, or a parallel implementation.

In Publications 3, 4, and 5 a massively parallel CNAPS neurocomputer was used to perform part of the SOM computations.

It has been proposed (Chang and Lee, 1973) that the computational complexity of Sammon's mapping could be reduced by applying it only to a representative subset of the total data set. The rest of the data items could then be mapped one by one, thereby only trying to optimize their distances from the fixed subset. This method is not as accurate as the original, however.

Besides the computational complexity, the existence of local minima in the cost functions may also cause problems. On the basis of empirical experience Sammon's mapping can easily get stuck at local optima (cf. Mao and Jain, 1995, and Ripley, 1996), and in practice the computation of the projection must be repeated several times starting from different initial configurations. The SOM algorithm, on the other hand, seems to be quite robust if the learning is begun with a wide neighborhood, which then shrinks gradually to its final narrow form (Kohonen, 1995c). There may be some differences in the resulting maps, depending on their initial state and the stochastic input sequence, however, and the effect of the differences in the actual use of the maps should be quantified empirically. The measures proposed in Publication 7 might thereby be of help.

Combinations of the methods. Since the different methods display different properties of the data set, the most useful approach is probably to use several of them together. An especially useful combination seems to be first to reduce the amount of data either by clustering or by the SOM, and then to display the reference vectors with some distance-preserving projection method to gain additional insight. In his original paper Sammon suggested that clustering could be used as a front-end to his mapping algorithm (Sammon, Jr., 1969). If the SOM is used to perform the clustering, there will be two different views to the same data available, which would certainly be useful.

It has also been proposed (Demartines, 1994; Demartines and Hérault, 1997) that in a combination of some vector quantization algorithm followed by metric MDS (or Sammon's mapping), the cost function of the latter could be modified slightly by introducing a decreasing weighting function F:

$$E_{M'} = \sum_{k \neq l} [d(k,l) - d'(k,l)]^2 F(d'(k,l)) \ . \tag{10}$$

The function F forces the mapping to concentrate on local distances.

3 STAGES OF SOM-BASED EXPLORATORY DATA ANALYSIS

In this section guidelines are presented for using the SOM in practical applications such as the case studies in the Publications belonging to this thesis. Related treatments have been presented earlier (Kaski and Kohonen, 1995; Kohonen, 1995c).

The basic steps in applying the SOM are quite standard although, as usual in exploratory data analysis, care must be taken in choosing the inputs and in evaluating the results to avoid misleading conclusions. The steps of the analysis can be used interactively by modifying the composition of the data set on the basis of insight gained during the previous round.

The first step in the analysis consists of the choice of the data set; although this step seems evident it cannot be stressed too much. No matter how good the methodology is the results will fundamentally depend on the quality and suitability of the data. Even good methods may only aid in the process of gaining insight into the nature of the data.

3.1 Preprocessing

Feature extraction, the choice of suitable representations for the data items, is a key step in the analysis. All unsupervised methods merely illustrate some structures in the data set, and the structures are ultimately determined by the features chosen to represent the data items. The usefulness of different preprocessing methods depends strongly on the application. Therefore a comprehensive treatment would be an immense task, and only the basic approaches used in the case studies can be introduced here.

If the data comes from a process of which there exists some a priori knowledge, this information should of course be used for choosing the features. This is the case with the EEG signal (Publication 1); it is known that the frequency content of the signal depends, for instance, on the vigilance of the individual. The better the features can be tailored to reflect the requirements of the task the better the results will be. In general, however, the task of tailoring requires considerable expertise both in the application area and in the data analysis methodology. Some experiments with a method that is potentially useful as an automatic feature extraction stage are reported in Publication 8.

Sophisticated feature extraction methods are also required in the case of mining symbolic information like text, for which no evident automatically available semantic features exist. Similarity relations computed based on the forms of the words would convey almost no information about the meaning and use of the words. *Contextual information* can, however, be used for constructing useful similarity relationships for textual data (Ritter and Kohonen, 1989). A system that utilizes representations of the "average context" of words to represent the words, and suitably processed *word category histograms* to represent documents is discussed in Publications 3, 4, 5, and 6.

Besides the choice of the features, their *scaling* must also be chosen before applying the SOM algorithm. If there exists knowledge of the relative importance of the components of the data items, the corresponding dimensions of the input space can be scaled according to this information. The importance may in some applications be estimated automatically by, for example, an entropy-based criterion (Publication 4). If no such criterion is available the variance of all components may be scaled to an equal value as was done in Publication 2.

The importance of choosing a set of indicators that describes the phenomenon of interest and nothing else, and proper scaling of the indicators cannot be over-stressed. Even the best analysis methods cannot overcome all mistakes made at this stage.

3.2 Computation of the maps

Detailed guidelines for how to actually compute the maps are given by Kohonen (1995c), as well as in the documentation of the public domain program package SOM_PAK (Kohonen et al., 1996a). The reference vectors are first initialized to lie in an ordered configuration on the plane spanned by the two principal eigenvectors of the data, and thereafter taught in a two-phase process. The learning starts with a wide neighborhood kernel covering most of the map, and during the first phase the kernel quickly narrows close to its final width, at the same time becoming smaller in its peak amplitude. During the second, longer phase, the neighborhood kernel continues from the narrower form and slowly shrinks to its final width and magnitude. The first phase enforces a global ordering of the map, while in the second phase the final accurate state of the map is formed gradually. The final neighborhood width determines the "stiffness" of the map, i.e., how closely the map will follow the local structures in the data.

3.3 Choosing good maps

Since the SOM learning process described in Section 2.4.1 is stochastic, there will be some variation left in the learning results. Thus, to ensure good quality several maps can be computed and the best map can be chosen according to the cost function, Equation 7.

The cost function of the SOM is specific to the size of the map and to the topology of the map lattice, which is defined by the neighborhood kernel. The value of the cost function will generally decrease as the map size increases, and increase as the width of the neighborhood function increases. Thus, it cannot be used to compare maps with different sizes or neighborhood kernels; some auxiliary criteria are therefore needed.

A measure of how well a map preserves the topology of the input space might be a suitable auxiliary criterion of map goodness. Topology preservation has, however, turned out to be quite difficult to define sensibly for a discrete grid. There seem to exist two different approaches for measuring the degree of topology preservation (specific examples have been reviewed in Publication 7; additional approaches have been proposed at least by Zrehen, 1993; and Hämäläinen, 1994).

In the first approach the relations between the reference vectors and the relations between the corresponding units on the map lattice are compared. For example, it can be measured how often the line connecting the reference vectors of neighboring units will be dissected by the Voronoi region of some other unit than the endpoints (Zrehen, 1993). The Voronoi region of unit i is defined to be the set consisting of the points in the input space that are closer to $\mathbf{m}_i$ than to any other reference vector.

An alternative approach for measuring topology preservation is to use input samples to determine how *continuous* the mapping *from the input space to the map grid* is. If the two reference vectors that are closest to a given data item are not neighbors on the map lattice, then the mapping is not continuous near that item.

Neither of these approaches takes into account the accuracy of the map in representing the data, the first term in the decomposition of the cost function (Eq. 9). In Publication 7 it is proposed that the goodness of the SOM could be measured by using a sum of the accuracy and a suitably chosen index of the topology preservation (cf. Fig. 6).

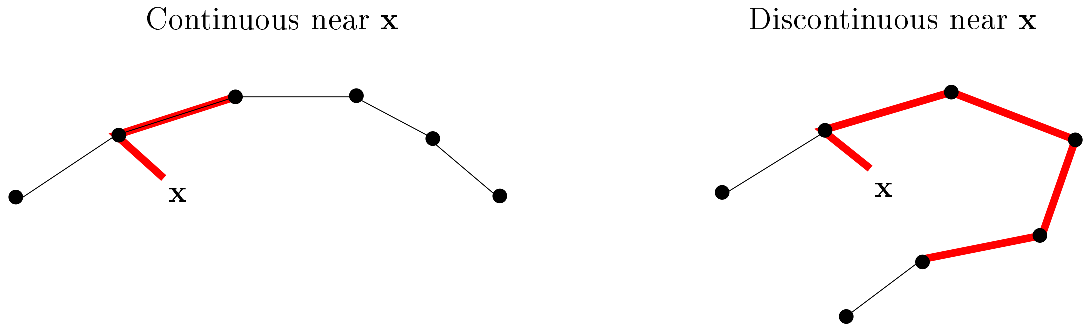

Figure 6: One method for measuring the goodness of SOMs consists of computing, for a representative set of input samples $\mathbf{x}$, the distance from $\mathbf{x}$ first to the closest reference vector, and thereafter to the second-closest reference vector along the map. The index of goodness is the average of these distances. If the mapping from the input space to the map grid is continuous near $\mathbf{x}$ the distances are generally smaller than when the mapping is discontinuous. The dots denote reference vectors of a one-dimensional map in a two-dimensional input space. Reference vectors of neighboring map units have been connected with thin lines. The thick line indicates how the distance is computed. For two-dimensional maps the distance is computed using the shortest path along the map.

A related measure has been proposed by Minamimoto et al. (1995) for analyzing the topological structure of the data space. There are two essential differences between the measures, however: first, Minamimoto et al. combine qualitatively different measures using a linear combination. The coefficients of the combination may then be difficult to choose, whereas in Publication 7 the measures that are combined measure distances in the same space. Second, Minamimoto et al. also combine the number of reference vectors in the measure. We, however, considered the size of the map only indirectly, to the extent that is contributes to the preservation of the topology.

The best way to use the measure of goodness for choosing a map, assuming there are enough computational resources, seems to be to teach a set of maps with each choice of the map size and the neighborhood kernel. The map that minimizes the cost function (Eq. 7) is then chosen from each set. Of the resulting set of best maps, the final map is chosen according to the measure of goodness presented in Publication 7.

Some additional notes, for which there was no space in Publication 7, are given below:

Note 1: Sparse data. If the data set is sparse, consisting only of a few samples per map unit, it becomes more difficult to measure topology preservation. A judgment of the continuity of the mapping based on a sum over the data samples may become inaccurate. It may then be useful to consider, along with the distance between the nearest and second-nearest reference vector as was done in Publication 7, the distance to the third-closest reference vector etc., weighted suitably.

Note 2: High dimensionality. One motivation for using an index of continuity of the mapping from the input space to the map grid as a measure of the goodness of the mapping stems from the fact that the lower-dimensional SOM grid tends to fold when trying to follow a distribution of a higher dimensionality. Discontinuities in the mapping could then indicate the presence of such folds. This motivation may be misleading for high-dimensional data spaces which are necessarily sparse because of the "curse of dimensionality" (cf. Bellman, 1961). There may be too few samples for distinguishing between a non-linear curve and a sample from a higher-dimensional distribution, for instance. The goodness measure proposed in Publication 7 may, however, be a useful index of the *regularity* of the mapping even in such cases, and if there are local lower-dimensional regions in the input space the measure will be useful in those regions.

Note 3: Computational complexity. The proposed measure is fairly computationally intensive. It requires searching for the shortest path between each pair of units on the map, based on the distances between the reference vectors in the data space. The measure can, however, be computed using dynamic programming (cf., e.g., Sedgewick, 1988), which reduces the complexity. A coarse estimate of the measure, suitable for very large maps, might also be obtained by computing the distance along the reference vectors of the units that lie on the shortest path along the map lattice, or even by computing the distance along the map lattice.

3.4 Interpretation, evaluation, and use of the maps

Interpretation. The interpretation of the findings of exploratory data analysis depends, of course, on the application. There exist, however, some general methods that may aid in the interpretation process.

Some caution is due at the very beginning of the interpretation: although the map metaphor may be useful for intuitively understanding what kinds of applications would be worthwhile, it does not necessarily hold all the way through to the interpretation of the map. Road maps, for example, basically only *scale* the distances, but the SOM may transform the locations of the data items in a highly nonlinear manner. Therefore it is not sensible to try to interpret the vertical and horizontal axes of the map in general, although in some special cases as in Publication 2 there may exist straightforward interpretations. If desired, simple interpretations may be sought by displaying auxiliary information on the map display and by inspecting its distribution. In Publication 2 the longer axis of the map seems to correlate with the overall economical welfare as measured by the GNP (gross national product) per capita.

Since the SOM tries above all to preserve local structures, the interpretation of the map should predominantly be done locally, based on the local relations of the data items on the map. The global structure is often useful as well, however. Different properties of the reference vectors and of the data items can be visualized on the map display to aid in the interpretation, as was discussed in Section 2.4.2.

If the data items come from a time-varying process, it is possible to visualize the trajectory of the successive samples on the map, and thereby to monitor the state of the process on an easily understandable visual display (cf., e.g., Alander et al., 1991; Kangas, 1994; Kasslin et al., 1992; Kohonen, 1995c; Tryba and Goser, 1991). Such trajectory displays are used in Publication 1.

Yet another method that aids in the interpretation of the maps, provided that some external information like class labels is available, is to plot the labels on the organized map. The distribution of the samples of each class, plotted on the map as a density histogram, may also help in the interpretation process. Distributions of samples containing different types of background EEG activity have been displayed in Publication 1, and different discussion topics (Usenet newsgroups) in Publication 4.

If the distributions of the known classes are overlapping such displays can even be used to explore the degree of overlap in different types of samples, whereby it may be possible to gain insight into whether the classes actually co-exist or whether new kinds of features should be added to the data items to make the classes more easily separable.

Displays of the reference vectors may also be useful. The methods for visualizing high-dimensional data discussed in Section 2.1 could be used, as well as some application-specific visualization methods like the head-shaped displays in Publication 1.

Evaluation. The quality of the map display should generally be evaluated by an expert in the application area. If samples having known classes are available, it is potentially useful to try to classify the samples using the map. Each map unit is labeled according to a majority voting of the samples, after which all samples that are projected into a unit are classified according to its label. The classification accuracy then indicates how well the classes are separated on the map, and the classification accuracy of new samples measures the generalizability of the result. Classification accuracy was used in this manner in Publications 1, 4, and 6.

The generalizability of the results could also give some indication of the quality of the mapping. Generalizability could be measured as the sensitivity of the map to small variations in the input data, caused either by adding artificial noise or by cross-validation. For the purposes of exploratory data analysis the sensitivity of the visualization should be measured instead of the sensitivity of the reference vectors of the SOM as such. In Publication 7 one possible sensitivity measure is presented; it reflects the difference in how two maps *represent visually* the relations between pairs of data items.

Use of the organized maps for exploratory data analysis. The illustrations formed by the SOM can be used as such, as tools for gaining insight into a data set. They can also be used to summarize data sets, together with the results of explorative research, or even as a decision-support system (DSS) (cf. Serrano-Cinca, 1996).

The SOM can be used in facilitating exploration of a data set, searching for known kinds of data, filtering of new incoming data, as well as visualization of the results.

The SOM can also be used for extracting clusters automatically (Lampinen and Oja, 1992; Murtagh, 1995; Pedrycz and Card, 1992; Varfis and Versino, 1992), and for rule extraction (Ultsch, 1993a).

4 CASE STUDIES

This thesis consists of three case studies in addition to some methodological developments. Although the case studies serve here as demonstrations of the use of the SOM in three different kinds of analysis tasks, an equally important motivation behind the studies has been the need for the actual applications themselves.

4.1 Multichannel EEG signal

The electroencephalogram (EEG) (Lopes da Silva et al., 1986; Niedermeyer and Lopes da Silva, 1987; Nunez, 1981) consists of a set of signals measured with electrodes on the scalp. The pattern of changes in the signals reflects some large-scale brain activity; for example the occurrence of certain kinds of oscillation patterns is known to be correlated with certain vigilance states of the subject. In addition to brain activity, the EEG also reflects activation of the head musculature, eye movements, interference from nearby electric devices, and changing conductivity in the electrodes due to the movements of the subject or physiochemical reactions at the electrode sites. All of these activities that are not directly related to the current cognitive processing of the subject are collectively referred to as *background activity* below.

EEG measurements provide plenty of continuous-valued, time-dependent data, and an EEG-specific feature extraction procedure is needed for revealing interesting activity patterns. In Publication 1 the background EEG activity is visualized using the SOM, after extracting frequency-based features from the short-time spectra of the EEG signals of all channels. The 140-dimensional data items correspond to the short-time frequency content of the EEG in multiple bands and in multiple locations on the scalp.

The resulting map distinguished between different kinds of background activity. The different activity types were predominantly projected onto different areas of the map. Each of the areas was more or less connected, even though samples from many subjects were used. By visualizing the trajectory of successive data items it is possible to monitor changes in the background activity, and the types of activity underlying different map areas can be inspected by visualizing the reference vectors.

4.2 Statistical tables

The *standard of living* involves a wealth of different aspects ranging from health and education to the quality of the environment. It is a very time-consuming task to acquire insight into a statistical table of indicators that represents the numerous relevant aspects. Such a task is then a precisely suitable application for exploratory data analysis methods.

In Publication 2 a total of 39 indicators describing different aspects of the standard of living were chosen from a World Development Indicator set (World Bank, 1992), and a display of the *structures of welfare and poverty* that the data set reveals was set up using the SOM. The dominant axis on the map was found to be correlated with the GNP per capita, which was not included into the teaching corpus, by displaying the distribution of the GNP per capita values on the groundwork formed of the organized map. The interpretation of the fine structures of welfare and poverty can be done by inspecting the distributions of the values of the original indicators on the map groundwork.

In this study the input data set was finite, and the components of the data items were scaled to have equal variance, since it would have been difficult to determine differences in their importance. In the study it was also demonstrated how the missing value problem can be treated.

4.3 Full-text document collections

A project that aims at constructing methods for exploring full-text document collections, the WEBSOM, started from Timo Honkela's suggestion of using the "self-organizing semantic maps" (Ritter and Kohonen, 1989) as a preprocessing stage for encoding documents. When the documents are organized, using this preprocessing stage, on a map in such a way that nearby locations contain similar documents, exploration of the collection is facilitated by the intuitive neighborhood relations. Structures in the collection can be visualized with the methods described in Section 2.4.2.

The basic method is described in Publication 3; experiments with very large maps and document collections are in Publication 4; and the browsing interface and exploration examples are in Publication 5. A partly supervised version of the method has also been constructed (Honkela et al., 1996). The maps that have been presented in the publications are available for exploration in the Internet at the address `http://websom.hut.fi/websom/`.

The advantages gained by using such a SOM-based feature extraction stage in WEBSOM are analyzed in more detail in Publication 6. It has turned out that the self-organizing semantic map can be used to form a computationally efficient approximation of a probabilistic model that takes into account contextual information in encoding the documents.

4.3.1 Recent developments

It has turned out quite recently that a simpler document encoding method than the one that was used in the Publications might produce even better results.

It is a standard practice in information retrieval (IR) (Salton and McGill, 1983) to encode documents with vectors, in which each component corresponds to a different word, and the value of the component reflects the frequency of occurrence of the word in the document. If word s_k occurs $n_k^{(j)}$ times in document j, $p_k^{(j)}$ is defined to be equal to $n_k^{(j)} / \sum_{k'} n_{k'}^{(j)}$, and $\mathbf{e}_k$ is the unit vector corresponding to the kth vector component, it is possible to code the document j by

$$\mathbf{a}^{(j)} = \sum_k p_k^{(j)} \mathbf{e}_k \ . \tag{11}$$

A problem with this encoding method is that if the vocabulary is very large the dimensionality of the vector is also high. In the Publications this problem was solved by eliminating some of the most common and some of the rarest words, and by clustering the words into word categories. (Actually an extended version of Equation 11 was used, where probabilistic information about the similarity of use of different words was incorporated into the coefficients $p_k^{(j)}$.)

An alternative approach for reducing the dimensionality is simply to reduce the dimensionality of the vectors $\mathbf{e}_k$ that in effect represent the words. A simple method is to project them randomly to a space of a lower dimensionality (Ritter and Kohonen, 1989). It has turned out in experiments that using the reduced-dimensional version of Equation 11 without any contextual information results in better separability of different Usenet discussion areas (accuracy of the newsgroup separation was 69.1 %) than using either the current WEBSOM method (accuracy 62.6 %), or a method where vectors estimated based on contextual information are used in the place of the $\mathbf{e}_k$ (Gallant et al., 1992; Gallant, 1994) (accuracy 65.4 %). The experimental procedures have been described in more detail in Publication 6. If this reduced-dimensional version of Equation 11 is used, however, the fast processing of documents by table lookups and subsequent convolutions, used in Publications 3, 4, 5, and 6, would become impossible.

These preliminary results need more thorough validation. It may in any case be concluded that the methods for utilizing contextual information can still be improved.

5 FURTHER DEVELOPMENTS

5.1 Feature exploration with the adaptive-subspace SOM

Feature extraction is perhaps the most difficult task of the whole enterprise of data exploration. The components of the data vectors should be selected and preprocessed so that the relations between the representations of the data items would correspond to meaningful relations between the items. Considerable expertise in the application area may be required for building suitable feature extractors.

The adaptive-subspace SOM (ASSOM) (Kohonen, 1995a; Kohonen, 1995b; Kohonen, 1995c; Kohonen, 1996) is a step towards a more general-purpose feature extractor: it extracts *invariant features* from its input, features that are invariant to the particular transformation that has operated on the inputs.

The ASSOM could act as a learning feature extraction stage that produces invariant representations to be explored by the SOM, or in *feature exploration.* The ASSOM extracts the invariant features with filters that are formed automatically based on short sequences of input samples during the learning process. Insight into the processes that have produced the data set might then be gained by visualizing the resulting filters. The visualization is particularly easy since the representations of the ASSOM are ordered just as in the basic SOM.

In Publication 8 the study of the ASSOM algorithm is continued; both the theoretical treatment and the scope of the experimental simulations are broadened. It is demonstrated that features invariant to different kinds of transformations can be extracted, and that areas in which all units have been specialized to be invariant to one of the transformations emerge if several transformations have been present in the training data, albeit at different times.

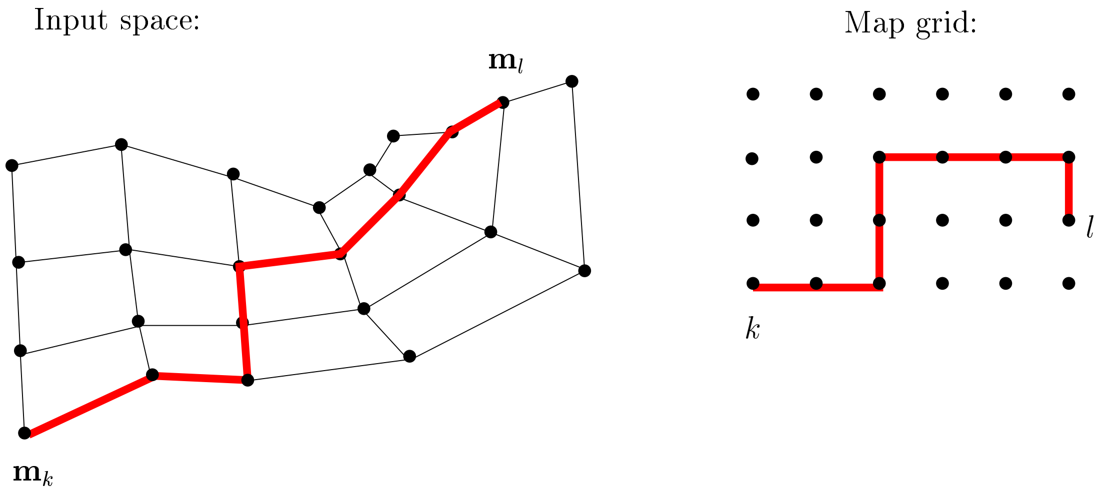

Figure 7: The nonlinearity of the SOM is taken into account by defining the distance between map units to be the distance along the "elastic network" formed of the map in the input space. On the left the reference vectors of a two-dimensional map in a two-dimensional input space have been denoted by dots. Neighboring reference vectors have been connected with thin lines. Distance between units k and l is drawn with the thick line. Depending on the values of the reference vectors the path along which the distance is computed need not be the shortest path on the map grid, as shown on the right.

5.2 Comparison of knowledge areas

If SOMs were adopted on a large scale for summarizing information in various data sets, it might be of use to be able to compare the data sets indirectly by comparing the "summaries" formed by the ordered sets of reference vectors. Possible application areas could include the comparison of organizational data ("data warehousing", making data on an organization or company available for on-line retrieval is nowadays quite popular, cf. Fayyad et al., 1996c) and comparison of the expertise of different parties for deciding what they could learn from each other.

Such comparisons of different maps should focus on the "equivalence of use", or similarity of the representations of knowledge the maps form. A measure of the similarity of two maps based on how they represent *relations* between data items has been presented in Publication 7. The relation, here the *distance*, between the representations of two data items *on the map display* is computed taking into account the nonlinearity of the map: the distance between each pair of neighboring map units is first defined to be the distance of the corresponding reference vectors. The distance of any two map units is defined to be the distance along the minimum path from one of the units to the other, along the map (cf. Kraaijveld et al., 1992; Kraaijveld et al., 1995). The computation of the distance is illustrated in Figure 7. The distance between any two data items is then defined to be the sum of the distances from each of the data items to the closest reference vector, plus the distance between the corresponding units on the map.

The measure of the (dis)similarity of two maps, a *metric* for SOMs, can then be constructed by comparing the relations of pairs of data items on the two maps.

6 CONCLUSION

In this thesis methodologies have been established for applying self-organizing maps in the exploratory analysis of large data sets. The methods have been demonstrated in three different kinds of case studies: continuous-valued dense data (EEG signals), continuous-valued sparse data (indicators of the standard of living of different countries), and discrete-valued data (full-text document collections).

The self-organizing maps illustrate structures in the data in a different manner than, for example, multidimensional scaling, a more traditional multivariate data analysis methodology. The SOM algorithm concentrates on preserving the neighborhood relations in the data instead of trying to preserve the distances between the data items. Comparisons between methods having different goals must eventually be based on their practical applicability. Here the SOM has been shown to provide a viable alternative.

REFERENCES

Alander, J. T., Frisk, M., Holmström, L., Hämäläinen, A., and Tuominen, J. (1991) Process error detection using self-organizing feature maps. In Kohonen, T., Mäkisara, K., Simula, O., and Kangas, J., editors, *Artificial Neural Networks. Proceedings of ICANN'91, International Conference on Artificial Neural Networks*, volume II, pages 1229–1232. North-Holland, Amsterdam.

Amari, S. (1980) Topographic organization of nerve fields. *Bulletin of Mathematical Biology*, 42:339–364.

Anderberg, M. R. (1973) *Cluster Analysis for Applications.* Academic Press, New York, NY.

Andrews, D. F. (1972) Plots of high-dimensional data. *Biometrics*, 28:125–136.

Angéniol, B., de La Croix Vaubois, G., and Le Texier, J.-Y. (1988). Self-organizing feature maps and the traveling salesman problem. *Neural Networks*, 1:289–293.

Back, B., Sere, K., and Vanharanta, H. (1996) Data mining accounting numbers using self-organizing maps. In Alander, J., Honkela, T., and Jakobsson, M., editors, *Proceedings of STeP'96, Finnish Artificial Intelligence Conference*, pages 35–47. Finnish Artificial Intelligence Society, Vaasa, Finland.

Bellman, R. E. (1961) *Adaptive Control Processes: A Guided Tour.* Princeton University Press, New Jersey, NJ.

Bezdek, J. C. and Pal, N. R. (1995) An index of topological preservation for feature extraction. *Pattern Recognition*, 28:381–391.

Bishop, C. M., Svensén, M., and Williams, C. K. I. (1996a). EM optimization of latent-variable models. In Touretzky, D. S., Mozer, M. C., and Hasselmo, M. E., editors, *Advances in Neural Information Processing Systems 8*, pages 465–471. The MIT Press, Cambridge, MA.

Bishop, C. M., Svensén, M., and Williams, C. K. I. (1996b). GTM: a principled alternative to the self-organizing map. In von der Malsburg, C., von Seelen, W., Vorbrüggen, J. C., and Sendhoff, B., editors, *Proceedings of ICANN'96, International Conference on Artificial Neural Networks*, Lecture Notes in Computer Science, vol. 1112, pages 165–170. Springer, Berlin.

Biswas, G., Jain, A. K., and Dubes, R. C. (1981) Evaluation of projection algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3:701–708.

Blackmore, J. and Miikkulainen, R. (1993) Incremental grid growing: encoding high-dimensional structure into a two-dimensional feature map. In *Proceedings of ICNN'93, IEEE International Conference on Neural Networks*, volume I, pages 450–455. IEEE Service Center, Piscataway, NJ.

Blayo, F. and Demartines, P. (1992) Algorithme de Kohonen: application à l'analyse de données économiques. *Bulletin des Schweizerischen Elektrotechnischen Vereins & des Verbandes Schweizerischer Elektrizitätswerke*, 83(5):23–26.

Bruske, J. and Sommer, G. (1995) Dynamic cell structure learns perfectly topology preserving map. *Neural Computation*, 7:845–865.

Budinich, M. (1996). A self-organizing neural network for the traveling salesman problem that is competitive with simulated annealing. *Neural Computation*, 8:416–424.

Carlson, E. (1991) Self-organizing feature maps for appraisal of land value of shore parcels. In Kohonen, T., Mäkisara, K., Simula, O., and Kangas, J., editors, *Artificial Neural Networks. Proceedings of ICANN'91, International Conference on Artificial Neural Networks*, volume II, pages 1309–1312, North-Holland, Amsterdam.

Chang, C. L. and Lee, R. C. T. (1973) A heuristic relaxation method for nonlinear mapping in cluster analysis. *IEEE Transactions on Systems, Man, and Cybernetics*, 3:197–200.

Cheng, G., Liu, X., and Wu, J. X. (1994) Interactive knowledge discovery through self-organizing feature maps. In *Proceedings of WCNN'94, World Congress on Neural Networks*, volume IV, pages 430–434. Lawrence Erlbaum, Hillsdale, NJ.

Chernoff, H. (1973) The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, 68:361–368.

Cichocki, A. and Unbehauen, R. (1993) *Neural Networks for Optimization and Signal Processing.* John Wiley, Chichester, England.

Cooley, W. W. and Lohnes, P. R. (1971) *Multivariate Data Analysis.* Wiley, New York, NY.

de Leeuw, J. and Heiser, W. (1982) Theory of multidimensional scaling. In Krishnaiah, P. R. and Kanal, L. N., editors, *Handbook of Statistics*, volume 2, pages 285–316. North-Holland, Amsterdam.

Demartines, P. (1994) *Analyse de données par réseaux de neurones auto-organisés (Data analysis through self-organized neural networks).* PhD thesis, Institut National Polytechnique de Grenoble, Grenoble, France.

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*, 8:148–154.

DeMers, D. and Cottrell, G. (1993) Non-linear dimension reduction. In Hanson, S. J., Cowan, J. D., and Giles, C. L., editors, *Advances in Neural Information Processing Systems 5*, pages 580–587, Morgan Kaufmann, San Mateo, CA.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977) Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39:1–38.

Devijver, P. A. and Kittler, J. (1982) *Pattern Recognition: A Statistical Approach.* Prentice Hall, Englewood Cliffs, NJ.

Didday, R. L. (1970) *The Simulation and Modeling of Distributed Information Processing in the Frog Visual System.* PhD thesis, Stanford University.

Didday, R. L. (1976) A model of visuomotor mechanisms in the frog optic tectum. *Mathematical Biosciences*, 30:169–180.

Dixon, J. K. (1979) Pattern recognition with partly missing data. *IEEE Transactions on Systems, Man, and Cybernetics*, 9:617–621.

du Toit, S. H. C., Steyn, A. G. W., and Stumpf, R. H. (1986) *Graphical Exploratory Data Analysis.* Springer-Verlag, New York, NY.

Erwin, E., Obermayer, K., and Schulten, K. (1992). Self-organizing maps: ordering, convergence properties and energy functions. *Biological Cybernetics*, 67:47–55.

Fayyad, U. M. (1996). Data mining and knowledge discovery: making sense out of data. *IEEE Expert*, October, pages 20–25.

Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996a) Knowledge discovery and data mining: towards a unifying framework. In Simoudis, E., Han, J., and Fayyad, U., editors, *Proceedings of KDD'96, Second International Conference on Knowledge Discovery & Data Mining*, pages 82–88. AAAI Press, Menlo Park, CA.

Fayyad, U. M., Piatetsky-Shapiro, G., and Smyth, P., editors (1996b) *Advances in Knowledge Discovery and Data Mining.* AAAI Press / MIT Press, Menlo Park, CA.

Fayyad, U. M., Piatetsky-Shapiro, G., and Smyth, P. (1996c) 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*, pages 1–34. AAAI Press / MIT Press, Menlo Park, CA.

Flexer, A. (1997) Limitations of self-organizing maps for vector quantization and multidimensional scaling. To appear in Mozer, M. C., Jordan, M. I., and Petsche, T., editors, *Advances in Neural Information Processing Systems 9.*

Forsyth, R., editor (1989) *Machine Learning: Principles and Techniques.* Chapman and Hall, London.

Friedman, J. H. (1987) Exploratory projection pursuit. *Journal of the American Statistical Association*, 82:249–266.

Friedman, J. H. and Tukey, J. W. (1974) A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, 23:881–890.

Fritzke, B. (1991) Let it grow - self-organizing feature maps with problem dependent cell structure. In Kohonen, T., Mäkisara, K., Simula, O., and Kangas, J., editors, *Artificial Neural Networks. Proceedings of ICANN'91, International Conference on Artificial Neural Networks*, volume I, pages 403–408, North-Holland, Amsterdam.

Fritzke, B. (1994) Growing cell structures—a self-organizing network for unsupervised and supervised learning. *Neural Networks*, 7:1441–1460.

Fu, K. S. (1974) *Syntactic Methods in Pattern Recognition.* Academic Press, New York, NY.

Fukunaga, K. (1972) *Introduction to Statistical Pattern Recognition.* Academic Press, New York, NY.

Fyfe, C. and Baddeley, R. (1995) Non-linear data structure extraction using simple Hebbian networks. *Biological Cybernetics*, 72:533–541.

Gallant, S. I. (1994) Methods for generating or revising context vectors for a plurality of word stems. U.S. Patent number 5,325,298.

Gallant, S. I., Caid, W. R., Carleton, J., Hecht-Nielsen, R., Pu Qing, K., and Sudbeck, D. (1992) HNC's MatchPlus system. *ACM SIGIR Forum*, 26(2):34–38.

Garavaglia, S. (1993) A self-organizing map applied to macro and micro analysis of data with dummy variables. In *Proceedings of WCNN'93, World Congress on Neural Networks*, pages 362–368. Lawrence Erlbaum and INNS Press, Hillsdale, NJ.

Garrido, L., Gaitan, V., Serra-Ricart, M., and Calbert, X. (1995) Use of multilayer feedforward neural nets as a display method for multidimensional distributions. *International Journal of Neural Systems*, 6:273–282.

Gersho, A. (1979) Asymptotically optimal block quantization. *IEEE Transactions on Information Theory*, 25:373–380.

Goodhill, G. J., Finch, S., and Sejnowski, T. J. (1995) Quantifying neighborhood preservation in topographic mappings. Technical Report INC-9505, Institute for Neural Computation, La Jolla, CA.

Gray, R. M. (1984) Vector quantization. *IEEE ASSP Magazine*, April, pages 4–29.

Grossberg, S. (1976) On the development of feature detectors in the visual cortex with applications to learning and reaction-diffusion systems. *Biological Cybernetics*, 21:145–159.

Hair, Jr., J. F., Anderson, R. E., Tatham, R. L., and Black, W. C. (1984) *Multivariate Data Analysis with Readings* (4th edition 1995). Prentice-Hall, Englewood Cliffs, NJ.

Hämäläinen, A. (1994) A measure of disorder for the self-organizing map. In *Proceedings of ICNN'94, IEEE International Conference on Neural Networks*, volume II, pages 659–664. IEEE Service Center, Piscataway, NJ.

Hartigan, J. (1975) *Clustering Algorithms*. Wiley, New York, NY.

Hastie, T. and Stuetzle, W. (1989) Principal curves. *Journal of the American Statistical Association*, 84:502–516.

Haykin, S. (1994) *Neural Networks. A Comprehensive Foundation*. Macmillan, New York, NY.

Hecht-Nielsen, R. (1995) Replicator neural networks for universal optimal source coding. *Science*, 269:1860–1863.

Hoaglin, D. C. (1982) Exploratory data analysis. In Kotz, S., Johnson, N. L., and Read, C. B., editors, *Encyclopedia of Statistical Sciences*, volume 2, pages 579–583. Wiley, New York.

Honkela, T., Kaski, S., Lagus, K., and Kohonen, T. (1996) Exploration of full-text databases with self-organizing maps. In *Proceedings of ICNN'96, IEEE International Conference on Neural Networks*, volume I, pages 56–61. IEEE Service Center, Piscataway, NJ.

Hotelling, H. (1933) Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441,498–520.

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, *Proceedings of the Conference on Artificial Intelligence Research in Finland*, pages 122–126. Finnish Artificial Intelligence Society, Helsinki, Finland.

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

Jardine, N. and Sibson, R. (1971) *Mathematical Taxonomy.* Wiley, London.

Kangas, J. (1994) *On the analysis of pattern sequences by self-organizing maps.* PhD thesis, Helsinki University of Technology, Espoo, Finland.

Kaski, S. and Kohonen, T. (1994) Winner-take-all networks for physiological models of competitive learning. *Neural Networks*, 7:973–984.

Kaski, S. and Kohonen, T. (1995) Structures of welfare and poverty in the world discovered by the self-organizing map. Technical Report A24, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland.

Kasslin, M., Kangas, J., and Simula, O. (1992) Process state monitoring using self-organizing maps. In Aleksander, I. and Taylor, J., editors, *Artificial Neural Networks, 2. Proceedings of ICANN'92, International Conference on Artificial Neural Networks*, pages 1531–1534, North-Holland, Amsterdam.

Kendall, M. (1975) *Multivariate Analysis.* Charles Griffin & Company, London.

Kohonen, T. (1981) Construction of similarity diagrams for phonemes by a self-organizing algorithm. Report TKK-F-A463, Helsinki University of Technology, Espoo, Finland.

Kohonen, T. (1982) Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69.

Kohonen, T. (1984) *Self-Organization and Associative Memory.* (3rd edition 1989). Springer, Berlin.

Kohonen, T. (1990) The Self-Organizing Map. *Proceedings of the IEEE*, 78:1464–1480.

Kohonen, T. (1991) Self-organizing maps: optimization approaches. In Kohonen, T., Mäkisara, K., Simula, O., and Kangas, J., editors, *Artificial Neural Networks. Proceedings of ICANN'91, International Conference on Artificial Neural Networks*, volume II, pages 981–990, North-Holland, Amsterdam.

Kohonen, T. (1993) Physiological interpretation of the self-organizing map algorithm. *Neural Networks*, 6:895–905.

Kohonen, T. (1995a) The adaptive-subspace SOM (ASSOM) and its use for the implementation of invariant feature detection. In Fogelman-Soulié, F. and Gallinari, P., editors, *Proceedings of ICANN'95, International Conference on Artificial Neural Networks*, volume 1, pages 3–10. EC2 & Cie, Paris.

Kohonen, T. (1995b) Emergence of invariant-feature detectors in self-organization. In Palaniswami, M., Attikiouzel, Y., Marks II, R. J., Fogel, D., and Fukuda, T., editors, *Computational intelligence. A dynamic system perspective*, pages 17–31. IEEE Press, New York, NY.

Kohonen, T. (1995c) *Self-Organizing Maps.* Springer, Berlin.

Kohonen, T. (1996) Emergence of invariant-feature detectors in the adaptive-subspace self-organizing map. *Biological Cybernetics*, 75:281–291.

Kohonen, T., Hynninen, J., Kangas, J., and Laaksonen, J. (1996a) SOM_PAK: the self-organizing map program package. Technical Report A31, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland.

Kohonen, T., Oja, E., Simula, O., Visa, A., and Kangas, J. (1996b). Engineering applications of the self-organizing map. *Proceedings of the IEEE*, 84:1358–1384.

Koikkalainen, P. (1994) Progress with the tree-structured self-organizing map. In Cohn, A. G., editor, *Proceedings of ECAI'94, 11th European Conference on Artificial Intelligence*, pages 211–215, Wiley, Chichester, England.

Koikkalainen, P. (1995) Fast deterministic self-organizing maps. In Fogelman-Soulié, F. and Gallinari, P., editors, *Proceedings of ICANN'95, International Conference on Artificial Neural Networks*, volume II, pages 63–68, EC2 & Cie, Paris.

Koikkalainen, P. and Oja, E. (1990) Self-organizing hierarchical feature maps. In *Proceedings of IJCNN'90 (San Diego), International Joint Conference on Neural Networks*, volume II, pages 279–284, IEEE Service Center, Piscataway, NJ.

Kraaijveld, M. A., Mao, J., and Jain, A. K. (1992) A non-linear projection method based on Kohonen's topology preserving maps. In *Proceedings of 11ICPR, 11th International Conference on Pattern Recognition*, pages 41–45, IEEE Computer Society Press, Los Alamitos, CA.

Kraaijveld, M. A., Mao, J., and Jain, A. K. (1995) A nonlinear projection method based on Kohonen's topology preserving maps. *IEEE Transactions on Neural Networks*, 6:548–559.

Kruskal, J. B. (1964) Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29:1–27.

Kruskal, J. B. and Wish, M. (1978) *Multidimensional Scaling.* Sage University Paper series on Quantitative Applications in the Social Sciences, number 07-011. Sage Publications, Newbury Park, CA.

Lampinen, J. and Oja, E. (1992) Clustering properties of hierarchical self-organizing maps. *Journal of Mathematical Imaging and Vision*, 2:261–272.

Langley, P. (1996) *Elements of Machine Learning.* Morgan Kaufmann, San Francisco, CA.

Lee, R. C. T., Slagle, J. R., and Blum, H. (1977) A triangulation method for the sequential mapping of points from N-space to two-space. *IEEE Transactions on Computers*, 26:288–292.

Lloyd, S. P. (1957). Least squares quantization in PCM. Unpublished memorandum, Bell Laboratories. (Published in *IEEE Transactions on Information Theory*, 28:129-137, 1982).

Lopes da Silva, F. H., Storm van Leeuwen, W., and Rémond, A., editors (1986) *Handbook of Electroencephalography and Clinical Neurophysiology. Volume 2: Clinical Applications of Computer Analysis of EEG and other Neurophysiological Signals.* Elsevier, Amsterdam.

Luttrell, P. S. (1988). Self-organizing multilayer topographic mappings. In *Proceedings of ICNN'88, IEEE International Conference on Neural Networks*, volume I, pages 93–100. IEEE Service Center, Piscataway, NJ.

Luttrell, S. P. (1989). Hierarchical vector quantization. *IEE Proceedings*, 136:405–413.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In Le Cam, L. M. and Neyman, J., editors, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. Volume I: Statistics*, pages 281–297. University of California Press, Berkeley and Los Angeles, CA.

Makhoul, J., Roucos, S., and Gish, H. (1985) Vector quantization in speech coding. *Proceedings of the IEEE*, 73:1551–1588.

Mao, J. and Jain, A. K. (1995) Artificial neural networks for feature extraction and multivariate data projection. *IEEE Transactions on Neural Networks*, 6:296–317.

Martín-del-Brío, B. and Serrano-Cinca, C. (1993) Self-organizing neural networks for the analysis and representation of data: some financial cases. *Neural Computing & Applications*, 1:193–206.

Martinetz, T. and Schulten, K. (1991) A "neural-gas" network learns topologies. In Kohonen, T., Mäkisara, K., Simula, O., and Kangas, J., editors, *Artificial Neural Networks. Proceedings of ICANN'91, International Conference on Artificial Neural Networks*, volume I, pages 397–402. North-Holland, Amsterdam.

Martinetz, T. and Schulten, K. (1994) Topology representing networks. *Neural Networks*, 7:507–522.

Marttinen, K. (1993) SOM in statistical analysis: supermarket customer profiling. In Bulsari, A. and Saxén, B., editors, *Proceedings of the Symposium on Neural Network Research in Finland*, pages 75–80. Finnish Artificial Intelligence Society, Turku, Finland.

Michalski, R. S., Carbonell, J., and Mitchell, T., editors (1983) *Machine Learning: An Artificial Intelligence Approach*. TIOGA Publishing Company, Palo Alto, CA.

Minamimoto, K., Ikeda, K., and Nakayama, K. (1995) Topology analysis of data space using self-organizing map. In *Proceedings of ICNN'95, IEEE International Conference on Neural Networks*, pages 789–794. IEEE Service Center, Piscataway, NJ.

Mulier, F. and Cherkassky, V. (1995) Self-organization as an iterative kernel smoothing process. *Neural Computation*, 7:1165–1177.

Murtagh, F. (1995) Interpreting the Kohonen self-organizing feature map using contiguity-constrained clustering. *Pattern Recognition Letters*, 16:399–408.

Nass, M. M. and Cooper, L. N. (1975) A theory for the development of feature detecting cells in visual cortex. *Biological Cybernetics*, 19:1–18.

Niedermeyer, E. and Lopes da Silva, F., editors (1987) *Electroencephalography: Basic Principles, Clinical Applications and Related Fields*. Urban & Schwarzenberg, Baltimore, second edition.

Nunez, P. L. (1981) *Electric Fields of the Brain. The Neurophysics of EEG*. Oxford University Press, New York, NY.

Oja, E. (1983) *Subspace Methods of Pattern Recognition*. Research Studies Press, Letchworth, England.

Oja, E. (1992) Principal components, minor components, and linear neural networks. *Neural Networks*, 5:927–935.

Pedrycz, W. and Card, H. C. (1992) Linguistic interpretation of self-organizing maps. In *IEEE International Conference on Fuzzy Systems*, pages 371–378. IEEE Service Center, Piscataway, NJ.

Pérez, R., Glass, L., and Shlaer, R. J. (1975) Development of specificity in cat visual cortex. *Journal of Mathematical Biology*, 1:275–288.

Ripley, B. D. (1996). *Pattern Recognition and Neural Networks.* Cambridge University Press, Cambridge, Great Britain.

Ritter, H. (1991) Asymptotic level density for a class of vector quantization processes. *IEEE Transactions on Neural Networks*, 2:173–175.

Ritter, H. and Kohonen, T. (1989) Self-organizing semantic maps. *Biological Cybernetics*, 61:241–254.

Ritter, H., Martinetz, T., and Schulten, K. (1992) *Neural Computation and Self-Organizing Maps: An Introduction.* Addison-Wesley, Reading, MA.

Ritter, H. and Schulten, K. (1988). Kohonen's self-organizing maps: exploring their computational capabilities. In *Proceedings of the ICNN'88, IEEE International Conference on Neural Networks*, volume I, pages 109–116. IEEE Service Center, Piscataway, NJ.

Rubner, J. and Tavan, P. (1989) A self-organizing network for principal component analysis. *Europhysics Letters*, 10:693–698.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986) Learning internal representations by error propagation. In Rumelhart, D. E., McClelland, J. L., and the PDP Research Group, editors, *Paralled Distributed Processing. Explorations in the Microstructure of Cognition. Volume 1: Foundations*, pages 318–362. The MIT Press, Cambridge, MA.

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

Samad, T. and Harp, S. A. (1992) Self-organization with partial data. *Network: Computation in Neural Systems*, 3:205–212.

Sammon, Jr., J. W. (1969) A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, 18:401–409.

Schalkoff, R. J. (1992) *Pattern Recognition: Statistical, Structural and Neural Approaches.* Wiley, New York, NY.

Sedgewick, R. (1988) *Algorithms.* Addison-Wesley, Reading, MA, 2nd edition.

Serrano-Cinca, C. (1996) Self-organizing neural networks for financial diagnosis. To appear in *Decision Support Systems.*

Shepard, R. N. (1962) The analysis of proximities: multidimensional scaling with an unknown distance function. *Psychometrika*, 27:125–140; 219–246.

Simoudis, E. (1996). Reality check for data mining. *IEEE Expert*, October, pages 26–33.

Sneath, P. H. A. and Sokal, R. R. (1973) *Numerical Taxonomy.* Freeman, San Francisco, CA.

Swindale, N. W. (1980) A model for the formation of ocular dominance stripes. *Proceedings of the Royal Society of London*, B, 208:243–264.

Therrien, C. W. (1989) *Decision, Estimation, and Classification. An Introduction to Pattern Recognition and Related Topics.* Wiley, New York, NY.

Torgerson, W. S. (1952) Multidimensional scaling: I. Theory and method. *Psychometrika*, 17:401–419.

Tryba, V. and Goser, K. (1991) Self-organizing feature maps for process control in chemistry. In *Artificial Neural Networks. Proceedings of ICANN'91, International Conference on Artificial Neural Networks*, volume I, pages 847–852, North-Holland, Amsterdam.

Tryon, R. C. and Bailey, D. E. (1973) *Cluster Analysis.* McGraw-Hill, New York, NY.

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

Ultsch, A. (1993a) Knowledge extraction from self-organizing neural networks. In Opitz, O., Lausen, B., and Klar, R., editors, *Information and Classification*, pages 301–306. Springer-Verlag, Berlin.

Ultsch, A. (1993b) Self-organizing neural networks for visualization and classification. In Opitz, O., Lausen, B., and Klar, R., editors, *Information and Classification*, pages 307–313. Springer-Verlag, Berlin.

Ultsch, A. and Siemon, H. P. (1990) Kohonen's self organizing feature maps for exploratory data analysis. In *Proceedings of ICNN'90, International Neural Network Conference*, pages 305–308, Kluwer, Dordrecht.

Varfis, A. and Versino, C. (1992) Clustering of socio-economic data with Kohonen maps. *Neural Network World*, 2:813–834.

Velleman, P. F. and Hoaglin, D. C. (1981) *Applications, Basics, and Computing of Exploratory Data Analysis.* Duxbury Press, Boston, MA.

von der Malsburg, C. (1973) Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik*, 14:85–100.

Webb, A. R. (1995). Multidimensional scaling by iterative majorization using radial basis functions. *Pattern Recognition*, 28:753–759.

Wish, M. and Carroll, J. D. (1982) Multidimensional scaling and its applications. In Krishnaiah, P. R. and Kanal, L. N., editors, *Handbook of Statistics*, volume 2, pages 317–345. North-Holland, Amsterdam.

World Bank (1992) *World Development Report 1992.* Oxford University Press, New York, NY.

Young, F. W. (1985) Multidimensional scaling. In Kotz, S., Johnson, N. L., and Read, C. B., editors, *Encyclopedia of Statistical Sciences*, volume 5, pages 649–659. Wiley, New York, NY.

Young, G. and Householder, A. S. (1938). Discussion of a set of points in terms of their mutual distances. *Psychometrika*, 3:19–22.

Zador, P. L. (1982). Asymptotic quantization error of continuous signals and the quantization dimension. *IEEE Transactions on Information Theory*, 28:139–149.

Zhang, X. and Li, Y. (1993) Self-organizing map as a new method for clustering and data analysis. In *Proceedings of IJCNN'93 (Nagoya), International Joint Conference on Neural Networks*, pages 2448–2451. IEEE Service Center, Piscataway, NJ.

Zrehen, S. (1993) Analyzing Kohonen maps with geometry. In Gielen, S. and Kappen, B., editors, *Proceedings of ICANN'93, International Conference on Artificial Neural Networks*, pages 609–612, Springer, London.

APPENDIX: KEY TO THE COUNTRY NAMES

AFG	Afghanistan
AGO	Angola
ALB	Albania
ARE	United Arab Emirates
ARG	Argentina
AUS	Australia
AUT	Austria
BDI	Burundi
BEL	Belgium
BEN	Benin
BGD	Bangladesh
BGR	Bulgaria
BOL	Bolivia
BRA	Brazil
BTN	Bhutan
BUR	Myanmar
BWA	Botswana
CAF	Central African Rep.
CAN	Canada
CHE	Switzerland
CHL	Chile
CHN	China
CIV	Cote d'Ivoire
CMR	Cameroon
COG	Congo
COL	Colombia
CRI	Costa Rica
CSK	Czechoslovakia
DEU	Germany
DNK	Denmark
DOM	Dominican Rep.
DZA	Algeria
ECU	Ecuador
EGY	Egypt, Arab Rep.
ESP	Spain
ETH	Ethiopia
FIN	Finland
FRA	France
GAB	Gabon
GBR	United Kingdom
GHA	Ghana
GIN	Guinea
GRC	Greece
GTM	Guatemala
HKG	Hong Kong
HND	Honduras
HTI	Haiti
HUN	Hungary
HVO	Burkina Faso
IDN	Indonesia
IND	India
IRL	Ireland
IRN	Iran, Islamic Rep.
IRQ	Iraq
ISR	Israel
ITA	Italy
JAM	Jamaica
JOR	Jordan
JPN	Japan
KEN	Kenya
KHM	Cambodia
KOR	Korea, Rep.
KWT	Kuwait
LAO	Lao PDR
LBN	Lebanon
LBR	Liberia
LBY	Libya
LKA	Sri Lanka
LSO	Lesotho
MAR	Morocco
MDG	Madagascar
MEX	Mexico
MLI	Mali
MNG	Mongolia
MOZ	Mozambique
MRT	Mauritania
MUS	Mauritius
MWI	Malawi
MYS	Malaysia
NAM	Namibia
NER	Niger
NGA	Nigeria
NIC	Nicaragua
NLD	Netherlands
NOR	Norway
NPL	Nepal
NZL	New Zealand
OAN	Taiwan, China
OMN	Oman
PAK	Pakistan
PAN	Panama
PER	Peru
PHL	Philippines
PNG	Papua New Guinea
POL	Poland
PRT	Portugal
PRY	Paraguay
ROM	Romania
RWA	Rwanda
SAU	Saudi Arabia
SDN	Sudan
SEN	Senegal
SGP	Singapore
SLE	Sierra Leone
SLV	El Salvador
SOM	Somalia
SWE	Sweden
SYR	Syrian Arab Rep.
TCD	Chad
TGO	Togo
THA	Thailand
TTO	Trinidad and Tobago
TUN	Tunisia
TUR	Turkey
TZA	Tanzania
UGA	Uganda
URY	Uruguay
USA	United States
VEN	Venezuela
VNM	Viet Nam
YEM	Yemen, Rep.
YUG	Yugoslavia
ZAF	South Africa
ZAR	Zaire
ZMB	Zambia
ZWE	Zimbabwe