QUINQUENNIAL REPORT

1994 – 1998

Neural Networks Research Centre &

Laboratory of Computer and Information Science

Helsinki University of Technology

P.O. Box 5400

FIN-02015 HUT, Finland

L. Koivisto, editor

Otaniemi, March 1999

Contents

Preface

The Neural Networks Research Centre (NNRC, neuroverkkojen tutkimusyksikkö) of Helsinki University of Technology (HUT) was established for the years 1994–98, and its funding comes from the Academy of Finland and HUT. It was selected as one of the centers of excellence (COE) in 1995, and its mandate was first extended up to the end of 1999. The COE status has later been approved for the years 2000–2005. The purpose of the NNRC is to pursue research in new information processing methods called the neural networks, but the major part of its research has concentrated on the theory and applications of the class of algorithms called the *Self-Organizing Map (SOM)* and its variations. This may have been justified, since the SOM was originally conceived here, and some time ago we listed 3343 publications on the SOM research from all over the world. It may be reasonable that the NNRC tries to keep the leading role in this extensive field of research.

Although the Neural Networks Research Centre is directly subordinated to the Senate of the University of Technology, it historically emerged from the Laboratory of Computer and Information Science (LCIS, informaatiotekniikan laboratorio), and there exist close ties between the two. The personnels of these laboratories cooperate in many ways, share common duties and use the same laboratories and facilities. Therefore it would be difficult to separate the researches of these two laboratories. A special research area of the LCIS has to be mentioned: new nonlinear estimation methods, in particular the Independent Component Analysis, where interesting neural-network implementations have been developed.

This report describes the research results of both laboratories in the years 1994–1998.

Espoo, March 22, 1999

Teuvo Kohonen
Academy Professor
Director,
Neural Networks Research Centre
Helsinki University of Technology

Erkki Oja
Professor
Director,
Laboratory of Computer and
Information Science
Helsinki University of Technology

Olli Simula
Professor
Laboratory of Computer and
Information Science
Helsinki University of Technology

Employees during 1994 – 1998

Neural Networks Research Centre

Teuvo Kohonen, Dr.Tech., academy professor, director of research centre
Adrian Flanagan, Ph.D., researcher
Tapio Hiltunen, Dipl.Eng., researcher
Timo Honkela, Ph.D., researcher
Jari Kangas, Dr.Tech., researcher
Samuel Kaski, Dr.Tech., researcher
Mikko Kurimo, Dr.Tech., researcher
Krista Lagus, Dipl.Eng., researcher
Harri Lappalainen, Dipl.Eng., researcher
Lea Leinonen, MScD, researcher
Ville Pulkki, Dipl.Eng., researcher
Jarkko Salojärvi, Dipl.Eng., researcher
Panu Somervuo, Dipl.Eng., researcher
Antti Ahonen, project assistant
Jukka Honkela, project assistant
Jussi Hynninen, project assistant
Janne Nikkilä, project assistant
Vesa Paatero, project assistant
Antti Saarela, project assistant
Vesa Siivola, project assistant
Antti Vainonen, project assistant
Jarkko Venna, project assistant
Leila Koivisto, secretary

Laboratory of Computer and Information Science

Erkki Oja, Dr.Tech., professor, director of laboratory
Olli Simula, Dr.Tech., professor
Juha Karhunen, Dr.Tech., professor
Ari Visa, Dr.Tech., docent, laboratory engineer
Esa Alhoniemi, Dipl.Eng., researcher
Xavier Giannakopoulos, M.Sc., researcher
Johan Himberg, Dipl.Eng., researcher
Jaakko Hollmén, Dipl.Eng., researcher
Karl-Arne Hovitie, Dipl.Eng., researcher
Jarmo Hurri, Dipl.Eng., researcher
Aapo Hyvärinen. Ph.D., researcher
Jukka Iivarinen, Dr.Tech., researcher
Jyrki Joutsensalo, Dr.Tech., researcher
Kimmo Kiviluoto, Dipl.Eng., researcher
Jorma Laaksonen, Dr.Tech., researcher
Mikko Mäkipää, Dipl.Eng., researcher
Petteri Pajunen, Dr.Tech., researcher

Markus Peura, Lic.Tech., researcher
Kimmo Raivio, Dr.Tech., researcher
Miki Sirola, Dipl.Eng., laboratory engineer
Haitao Tang, Dr.Tech., researcher
Kimmo Valkealahti, Dr.Tech., researcher
Juha Vesanto, Dipl.Eng., researcher
Ricardo Vigário, M.Sc., researcher
Liu-Yue Wang, Dr.Tech., researcher
Jussi Ahola, project assistant
Matti Aksela, project assistant
Razvan Cristescu, project assistant
Pekka Hippeläinen, project assistant
Patrik Hoyer, project assistant
Mika Inki, project assistant
Markus Koskela, project assistant
Jyrki Maaranen, project assistant
Simona Mălăroiu, project assistant
Juha Parhankangas, project assistant
Jukka Parviainen, project assistant
Henry Stenberg, project assistant
Jaakko Särelä, assistant
Vuokko Vuori, project assistant
Markku Ranta, laboratory technician
Tarja Pihamaa, laboratory secretary

Research Projects in the

Neural Networks Research Centre

1 The Self-Organizing Map (SOM)

Teuvo Kohonen

1.1 Introduction

The SOM is a new, effective software tool for the visualization of high-dimensional data. It implements an orderly mapping of a high-dimensional distribution onto a regular low-dimensional grid. Thereby it is able to convert complex, nonlinear statistical relationships between high-dimensional data items into simple geometric relationships on a low-dimensional display. As it compresses information while preserving the most important topological and metric relationships of the primary data items on the display, it may also be thought to produce some kind of abstractions. These two aspects, visualization and abstraction, can be utilized in a number of ways in complex tasks such as process analysis, machine perception, control, and communication.

The SOM usually consists of a two-dimensional regular grid of nodes. A *model* of some observation is associated with each node (cf. Fig. 1).

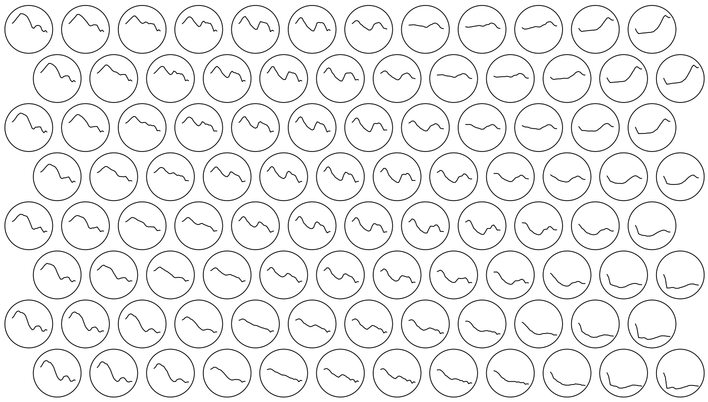


Figure 1: In this exemplary application, each processing element in the hexagonal grid holds a model of a short-time spectrum of natural speech (Finnish). Notice that neighboring models are mutually similar.

The SOM algorithm computes the models so that they optimally describe the domain of (discrete or continuously distributed) observations.

The models are automatically organized into a meaningful two-dimensional order in which similar models are closer to each other in the grid than the more dissimilar ones. In this sense the SOM is a similarity graph, and a clustering diagram, too. Its computation is a nonparametric, recursive regression process.

1.2 The incremental-learning SOM algorithm

Regression of an ordered set of model vectors $\mathbf{m}_i \in \Re^n$ into the space of observation vectors $\mathbf{x} \in \Re^n$ is often made by the following process:

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + h_{c(\mathbf{x}),i}(\mathbf{x}(t) - \mathbf{m}_i(t)) \ , \tag{1}$$

where t is the index of the regression step, and the regression is performed recursively for each presentation of a sample of $\mathbf{x}$, denoted $\mathbf{x}(t)$. The scalar multiplier $h_{c(\mathbf{x}),i}$ is called the *neighborhood function*, and it is like a smoothing or blurring kernel over the grid. Its first subscript $c = c(\mathbf{x})$ is defined by the condition

$$\forall i, \ \ \|\mathbf{x}(t) - \mathbf{m}_c(t)\| \leq \|\mathbf{x}(t) - \mathbf{m}_i(t)\| \ , \tag{2}$$

that is, $\mathbf{m}_c(t)$ is the model (called the *"winner"*) that matches best with $\mathbf{x}(t)$. The comparison metric is usually selected as Euclidean; for other metrics, the forms of (1) and (2) will change accordingly. If the samples $\mathbf{x}(t)$ are stochastic and have a continuous density function, the probability for having multiple minima in (2) is zero. With discrete-valued variables, multiple minima may occur; in such cases one of them should be selected at random for the winner.
The neighborhood function is often taken to be the Gaussian

$$h_{c(\mathbf{x}),i} = \alpha(t) \exp\left(-\frac{\|\mathbf{r}_i - \mathbf{r}_c\|^2}{2\sigma^2(t)}\right) \ , \tag{3}$$

where $0 < \alpha(t) < 1$ is the learning-rate factor, which decreases monotonically with the regression steps, $\mathbf{r}_i \in \Re^2$ and $\mathbf{r}_c \in \Re^2$ are the vectorial locations on the display grid, and $\sigma(t)$ corresponds to the width of the neighborhood function, which is also decreasing monotonically with the regression steps.
A simpler definition of $h_{c(\mathbf{x}),i}$ is the following: $h_{c(\mathbf{x}),i} = \alpha(t)$ if $\|\mathbf{r}_i - \mathbf{r}_c\|$ is smaller than a given radius from node c (whereupon this radius is a monotonically decreasing function of the regression steps, too), but otherwise $h_{c(\mathbf{x}),i} = 0$. In this case we shall call the set of nodes that lie within the given radius the *neighborhood set* N_c.
Due to the many stages in the development of the SOM method and its variations, there is often useless historical ballast in the computations.
For instance, an old ineffective principle is random initialization of the model vectors $\mathbf{m}_i$. Random initialization was originally used to show that there exists a strong self-organizing tendency in the SOM, so that the order can even emerge when starting from a completely unordered state, but this need not be demonstrated every time. On the contrary, if the initial values for the model vectors are selected as a regular array of vectorial values that lie on the subspace spanned by the eigenvectors corresponding to the two largest principal components of input data, computation of the SOM can be made orders of magnitude faster, since (i) the SOM is then already approximately organized in the beginning, (ii) one can start with a narrower neighborhood function and smaller learning-rate factor.
Many computational aspects like this and the selection of proper parameter values have been discussed in the software package SOM_PAK [1], as well as the book [2].

1.3 The batch version of the SOM

Another remark concerns faster algorithms. The incremental regression process defined by (1) and (2) can often be replaced by the following batch computation version which is significantly faster and does not require specification of any learning-rate factor $\alpha(t)$.
Assuming that the convergence to some ordered state is true, we require that the expectation values of $\mathbf{m}_i(t+1)$ and $\mathbf{m}_i(t)$ for $t \to \infty$ must be equal, even if $h_{ci}(t)$ were then selected nonzero. In other words, in the stationary state we must have

$$\forall i, \quad \mathrm{E}_t\{h_{c(\mathbf{x}),i}(\mathbf{x} - \mathbf{m}_i^*)\} = 0 \; . \tag{4}$$

In the special case where we have a finite number (batch) of the $\mathbf{x}(t)$ with respect to which (4) has to be solved for the $\mathbf{m}_i^*$, and $h_{c(\mathbf{x}),i}$ represents the kernels used during the last phases of the learning process, we can write (4) as

$$\mathbf{m}_i^* = \frac{\sum_t h_{c(\mathbf{x}),i}\mathbf{x}(t)}{\sum_t h_{c(\mathbf{x}),i}} \; . \tag{5}$$

This, however, is not yet an explicit solution for $\mathbf{m}_i^*$, because the subscript $c(\mathbf{x})$ on the right-hand side still depends on $\mathbf{x}(t)$ *and all the* $\mathbf{m}_i^*$. The way of writing (5), however, allows us to apply the *contractive mapping method* known from the theory of nonlinear equations: starting with even coarse approximations for the $\mathbf{m}_i^*$, (2) is first utilized to find the indices $c(\mathbf{x})$ for all the $\mathbf{x}(t)$. On the basis of the approximate $h_{c(\mathbf{x}),i}$ values, the improved approximations for the $\mathbf{m}_i^*$ are computed from (5), which are then applied to (2), whereafter the computed $c(\mathbf{x})$ are substituted to (5), and so on. The optimal solutions $\mathbf{m}_i^*$ are usually obtained in a few iteration cycles, after the discrete-valued indices $c(\mathbf{x})$ have settled down and are no longer changed in further iterations. This procedure is called the *Batch Map* principle.
An even simpler Batch Map principle is obtained if $h_{c(\mathbf{x}),i}$ is defined in terms of the neighborhood set N_c. Further we need the concept of the *Voronoi set*. It means a domain V_i in the $\mathbf{x}$ space, or actually the set of those samples $\mathbf{x}(t)$ that lie closest to $\mathbf{m}_i^*$. Let us recall that we defined N_i as the set of nodes that lie up to a certain radius from node i in the array. The union of Voronoi sets V_i corresponding to the nodes in N_i shall be denoted by U_i. Then (5) can be written

$$\mathbf{m}_i^* = \frac{\sum_{\mathbf{x}(t) \in U_i} \mathbf{x}(t)}{n(U_i)} \; , \tag{6}$$

where $n(U_i)$ means the number of samples $\mathbf{x}(t)$ that belong to U_i.
Notice again that the U_i depend on the $\mathbf{m}_i^*$, and therefore (6) must be solved iteratively. The procedure can be described as the following steps:

1. Initialize the values of the $\mathbf{m}_i^*$ in some proper way. (Even random values for the $\mathbf{m}_i^*$ will usually do.)

2. Input all the $\mathbf{x}(t)$, one at a time, and list each of them under the model $\mathbf{m}_i^*$ that is closest to $\mathbf{x}(t)$ according to (2).

3. Let U_i denote the union of the above lists at model $\mathbf{m}_i^*$ and its neighbors that constitute the neighborhood N_i. Compute the means of the vectors $\mathbf{x}(t)$ in each U_i, and replace the old values of $\mathbf{m}_i^*$ by the respective means.

4. Repeat from 2 a few times until the solutions can be regarded as steady.

A further acceleration of computation results if one notes that for the different nodes i, the same addends occur a great number of times. Therefore it is advisable to first compute the mean $\bar{\mathbf{x}}_j$ of the $\mathbf{x}(t)$ in each Voronoi set V_j and then weight it by the number n_j of samples in V_j and the neighborhood function. Now we obtain

$$\mathbf{m}_i^* = \frac{\sum_j n_j h_{ji} \bar{\mathbf{x}}_j}{\sum_j n_j h_{ji}} \,, \tag{7}$$

where the sum over j is taken for all units of the SOM. For the case in which neighborhood sets N_i are used,

$$\mathbf{m}_i^* = \frac{\sum_{j \in N_i} n_j \bar{\mathbf{x}}_j}{\sum_{j \in N_i} n_j} \,. \tag{8}$$

A convergence and ordering proof of the Batch Map has been presented in [3]. There is a Matlab SOM Toolbox program package available in the Internet at the address http://www.cis.hut.fi/projects/somtoolbox/, which makes use of the Batch Map method.

1.4 Learning Vector Quantization (LVQ)

If each of the sample vectors $\mathbf{x}(t)$ is known to belong to some predefined class, and the model vectors $\mathbf{m}_i(t)$ are labeled by symbols corresponding to the predefined classes, too, then a supervised-learning algorithm can be used to fine tune the model vectors [2]. The basic LVQ1 algorithm can be written in a compressed form as

$$\begin{aligned} \mathbf{m}_i(t+1) &= \mathbf{m}_i(t) + \alpha(t) s(t) \delta_{ci} [\mathbf{x}(t) - \mathbf{m}_i(t)] \,, \\[1em] \text{where } s(t) &= +1 \ \text{ if } \mathbf{x} \text{ and } \mathbf{m}_c \text{ belong to the same class,} \\ \text{but } s(t) &= -1 \ \text{ if } \mathbf{x} \text{ and } \mathbf{m}_c \text{ belong to different classes .} \end{aligned} \tag{9}$$

Here $\alpha(t)$ is the scalar-valued *learning-rate factor*, $0 < \alpha(t) < 1$, and δ_{ci} is the Kronecker delta ($= 1$ for $c = i$, $= 0$ for $c \neq i$) ; usually $\alpha(t)$ is initially of the order of a couple percent and decreases monotonically with time. The index c labels the winner according to (2). Notice that the neighborhood set around the winner now consists of the winner itself only.

Batch-LVQ1

The LVQ1 algorithm, like the SOM, can be expressed as a batch version. In a similar way as with the Batch Map (SOM) algorithm, the equilibrium condition for the LVQ1 is expresssed as

$$\forall i, \quad \mathrm{E}_t \left\{ s(t)\delta_{ci}(\mathbf{x} - \mathbf{m}_i^*) \right\} = 0 \; . \tag{10}$$

The computing steps of the so-called *Batch-LVQ1* algorithm (in which at steps 2 and 3 the class labels of the nodes are redefined dynamically) can then be expressed, in analogy with the Batch Map, as follows:

1. For the initial reference vectors take, for instance, those values obtained in the preceding unsupervised SOM process, where the classification of $\mathbf{x}(t)$ was not yet taken into account.

2. Input the $\mathbf{x}(t)$ again, this time listing the $\mathbf{x}(t)$ *as well as their class labels* under each of the corresponding winner nodes.

3. Determine the labels of the nodes according to the majorities of the class labels of the samples in these lists.

4. Multiply in each partial list all the $\mathbf{x}(t)$ by the corresponding factors $s(t)$ that indicate whether $\mathbf{x}(t)$ and $\mathbf{m}_c(t)$ belong to the same class or not.

5. At each node i, take for the new value of the reference vector the entity

$$\mathbf{m}_i^* = \frac{\sum_{t'} s(t')\mathbf{x}(t')}{\sum_{t'} s(t')} \; , \tag{11}$$

   where the summation is taken over the indices t' of those samples that were listed under node i.

6. Repeat from 2 a few times.

Comment 1. For stability reasons it may be necessary to check the sign of $\sum_{t'} s(t')$. If it becomes negative, no updating of this node is made.
Comment 2. Unlike in usual LVQ, the labeling of the nodes was allowed to change in the iterations. This has sometimes yielded slightly better classification accuracies than if the labels of the nodes were fixed at first steps. Alternatively, the labeling can be determined permanently immediately after the SOM process.

1.5 Further remarks

Finally it should be taken into account that the purpose of the SOM is usually visualization of data spaces. For an improved quality (isotropy) of the display it is then advisable to select the grid of the SOM units as hexagonal; the reason is similar as when using a hexagonal screen for images, say, in color television.
The above algorithms can be generalized, e.g., by defining various generalized matching criteria.
The following categories of similarity graphs, computed by the SOM, have already been used in many practical applications:

1. State diagrams for processes and machines

2. Data mining applications: similarity graphs for

   - statistical tables
   - full-text document collections

A list of 3043 research papers from very different application areas of the SOM and its variations is presented in [4].

References

[1] Kohonen, T., Hynninen, J., Kangas, J., Laaksonen, J. (1996). SOM_PAK: The self-organizing map program package. Report A31. Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland. Also available in the Internet at the address http://www.cis.hut.fi/nnrc/nnrc-programs.html.

[2] Kohonen, T. (1995). *Self-Organizing Maps*. Series in Information Sciences, Vol. 30. Springer, Heidelberg. Second ed. 1997.

[3] Cheng, Y. (1997). Convergence and ordering of Kohonen's batch map. *Neural Computation*, Vol. 9, pp. 1667-1676.

[4] Kangas, J. and Kaski, S. (1998) 3043 works that have been based on the self-organizing map (SOM) method developed by Kohonen. Report A50. Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland. Also available in the Internet at the address http://www.cis.hut.fi/nnrc/refs/references.ps.

2 Analyzing Self-Organization in the SOM

Adrian Flanagan

The SOM algorithm despite the simplicity of its implementation has shown itself to be particularly resistant to a general analysis of its self-organizing ability. For the most part theoretical analyses of the self-organizing property have been confined to the one dimensional case, that is a one dimensional input with a one-dimensional neuron grid. One of the reasons for this limitation is that it is only in this particular case that an organized state has been, so far, rigorously defined. Despite the robustness of the SOM which has been used successfully in many different application areas, very little is know from theory what conditions are sufficient for it to self-organize, and under what conditions it cannot organize. This project is concerned with extending already existing proofs of self-organization in the SOM, in a general way, such that sufficient conditions for self-organization to occur become apparent. To explain more specifically it is necessary to introduce some notation for the SOM. As so far the project has dealt with the one dimensional case, only the one dimensional SOM is described. The input $x \in \mathbb{R}$ is considered a random variable with probability distribution P. At each time iteration t a winner neuron c is chosen such that

$$c(t) \;=\; \arg\min_i |x(t) - m_i(t)| \tag{12}$$

where $m_i(t), i = 1, \ldots, N$ is the neuron weight value. Each neuron weight is then updated as follows,

$$m_i(t+1) \;=\; m_i(t) + \alpha(t)h(|i - c(t)|)(x(t) - m_i(t)). \tag{13}$$

The gain $\alpha(t)$, with $0 \leq \alpha(t) \leq 1$ during the training phase is normally a decreasing function with time. The function $h(|i - v(t)|)$ with $0 \leq h(|i - c(t)|) \leq 1$ is referred to as the *neighborhood* and $h(j)$ decreases with increasing j. In what follows $h(|i - c(t)|)$ will be written as $h(i, c(t))$. Generally h is defined as,

$$\begin{aligned} h(i,i) &= 1 \\ h(i, i \pm W) &= h_m > 0 \\ h(i,j) &= 0, \;\; |i-j| > W \\ h(i,j) &\leq h(i,k), \;\; |i-j| > |i-k| \end{aligned} \tag{14}$$

2.1 A Method for Analyzing Self-Organization

In the one dimensional case the organized configuration D of the neuron weights is *absorbing*,

$$D \;=\; \{\mathbf{M} : x_1 < x_2 < \ldots < x_N\} \bigcup \{\mathbf{M} : x_1 > x_2 > \ldots > x_N\} \tag{15}$$

and in Cottrell and Fort [2] it has been shown that from any initial condition where $m_i \neq m_j, i \neq j$, $W = 1$, and a uniform P that the weights will almost surely converge to D. This result was further generalized in Erwin et al [3], Bouton and

Pàges [4], Fort and Pàges [5], Flanagan [6], [7] and Sadeghi [8]. All of the latter consider $\mathbf{M}(t) = (m_1(t), m_2(t), \ldots,)$ as a Markov process defined on the common probability space $(\Psi, \mathcal{F}, \pi)$, and to prove self-organization it is shown that $\exists\, T < \infty$ and $\delta > 0$ for which

$$\pi_{\mathbf{M}(0)}(\{\psi \in \Psi \ : \ \tau_D \leq T\}) \ \geq \delta \tag{16}$$

or that the probability $\pi_{\mathbf{M}(0)}$, of finding sets of samples ψ in the sample space Ψ which take the neuron weights $\mathbf{M}$ from any initial condition $\mathbf{M}(0)$ to the organized configuration in a finite time τ_D is non zero. In [2], [3], [4], [5] and [8] either a uniform P or a diffuse P has been assumed. The generalization of these results is limited by the existence of situations where the inability to define a winner neuron can lead to the instability of the organized configuration. An example of when this may occur is $m_i(t) = m_j(t), i \neq j$ and

$$i, j \ = \ \arg \min_{1 \leq k \leq N} |x(t) - m_k(t)| \tag{17}$$

In [8] a modified version of the winner selection criterion of equation (12) to overcome this problem is presented along with a general analysis of the one dimensional SOM. This approach however is not generalizable to the higher dimensional case. In [6], [7] a different approach has been taken which avoids this problem of winner definition and it can be applied to both diffuse and discrete P and requires no change to the original SOM algorithm. The only restriction is that the neighborhood function $h(j)$ be assumed strictly monotonic decreasing with increasing j, that is

$$h(i,j) \leq h(i,k) - \phi \text{ for } |i-j| > |i-k|, \ \phi > 0 \tag{18}$$

In [6] it was shown that when $N \leq W$, for self-organization of the weights, the requirements on P are that its support contains a *skeleton structure* of two intervals, with $\int dP(x) > 0$ for each interval and that each interval be separated from the other by a certain minimum distance defined in terms of parameters of the map. The condition on P can be used both for discrete and diffuse P and this proof has already been easily extended to higher dimensional SOMs [6], which suggests the general framework of the proof developed in this project is not restricted to the one dimensional case. Define the *order*, n of an SOM as

$$n = \begin{cases} [\frac{N}{W}] + 1, & N \bmod W \neq 0 \\ \frac{N}{W}, & N \bmod W = 0 \end{cases} \tag{19}$$

and in this project the results of [6] (i.e. $n = 1$), and [7] (i.e. $n = 2$) are generalized, for the one dimensional case, for any $n \geq 1$. In other words general conditions that the SOM and support of P must satisfy are described and it is then proven that these conditions are sufficient for self-organization of the neuron weights for any $n \geq 1$ and any initial state of the neuron weights.

2.2 A Structure for Self-Organization

In the course of the project a structure has been defined, which if it exists in the support of P then self-organization can be shown. This structure $\mathcal{A}_n$, associated with an SOM of degree n, is defined in terms of two structures $\mathcal{A}_{n-1}$, separated by a certain minimum distance which depend on parameters of the SOM. Hence the structure $\mathcal{A}_n$ is defined recursively from 2^n basic structures $\mathcal{A}_0$, which is quite basically an interval on the line. Given this structure $\mathcal{A}_n$ with an SOM of degree n, then assuming that

$$\int_{\mathcal{A}_0} dP(x) > 0 \tag{20}$$

then the following theorem can be stated and proved.

Theorem 1 *For any initial, finite* $\mathbf{M}(0)$ *and the structure* $\mathcal{A}_n$ *such that* $N \leq nW$ *with*

$$\int_{\mathcal{A}_0} dP(x) \;\geq\; \epsilon, \qquad \epsilon > 0 \tag{21}$$

for every $\mathcal{A}_0$ *interval in* $\mathcal{A}_n$*, then* $\exists\, T < \infty$ *and* $\delta > 0$ *for which*

$$\pi_{\mathbf{M}(0)}(\{\psi \in \Psi \;:\; \tau_D \leq T\}) \;\geq \delta \tag{22}$$

where τ_D *is the first entry time of* $\mathbf{M}$ *into* D.

The recursive nature of the structure $\mathcal{A}_n$ leads to a proof by induction of the theorem. Throughout the proof three basic principles which apply at the level of the neuron weight updates are used, they are referred to as, convergence, order preservation and one step organization, and as well as being used in the one dimensional case, similar principles have been applied to higher dimensional SOMs.

2.3 Conclusion

By defining a special structure $\mathcal{A}_n$ on the support of P a general proof of self-organization in a one dimensional SOM has been given. The proof itself is theoretical, which raises many interesting questions concerning the implications of the proof in a practical situation. The conditions as determined are sufficient for self-organization to occur, but are they necessary in a practical situation, if not, how close are they to being necessary ? This question is very difficult from a theoretical point of view, given that the system being dealt with is stochastic. An estimation only of the importance of the conditions can be obtained from simulations.
Another interesting point of the structure $\mathcal{A}_n$ is the fact that it is self-similar, that is it looks the same on a large or small scale. In [9] the existence of a $1/f$ spectrum for the update of the neuron weights during training was shown by simulation. This is interesting in that there is a more general class of non linear systems which are referred to as *emergent*, they are usually associated with some form of self-similarity and a $1/f$ spectrum of some form. The significance if any of this relative to the SOM remains to be seen.

References

[1] Teuvo Kohonen. *Self-Organizing Maps*. Springer, Berlin, Heidelberg, 1995. (Second Extended Edition 1997).

[2] Marie Cottrell and Jean-Claude Fort. Étude d'un processus d'auto-organisation. *Annales de l'Institut Henri Poincaré*, 23(1):1–20, 1987. (in French).

[3] Ed Erwin, Klaus Obermayer, and Klaus Schulten. Self-organizing maps: Ordering, convergence properties and energy functions. *Biol. Cyb.*, 67(1):47–55, 1992.

[4] Catherine Bouton and Gilles Pagès. Self-organization and a. s. convergence of the one-dimensional Kohonen algorithm with non-uniformly distributed stimuli. *Stochastic Processes and Their Applications*, 47:249–274, 1993.

[5] Jean-Claude Fort and Gilles Pagès. On the a.s. convergence of the Kohonen algorithm with a general neighbourhood function. *Annals of Applied Probability*, 5(4):1177–1216, 1995.

[6] John A. Flanagan. Self-organization in Kohonen's SOM. *Neural Networks*, 9:1185–1197, 1996.

[7] John A. Flanagan. Sufficient conditions for self-organisation in the one dimensional SOM with a reduced width neighbourhood. *Neurocomputing*, 21:51–60, 1998.

[8] A.A. Sadeghi. Self-organisation and convergence of the one dimensional Kohonen algorithm. In *Proceedings ESANN98*, pages 173–178, Brussels, 1998. D Facto ed.

[9] John A. Flanagan. The self-organising map, robustness, self-organising criticality and power laws. In *Proceedings ESANN98*, pages 209–214, Brussels, 1998. D Facto ed.

3 Point Density of the Model Vectors in the SOM

Teuvo Kohonen

3.1 Introduction

In the classical vector quantization (VQ) the objective is usually to approximate n-dimensional real signal vectors $\mathbf{x} \in \mathbb{R}^n$ using a finite number of quantized vectorial values $\mathbf{m}_i \in \mathbb{R}^n, i = 1, \ldots, N$ called the codebook vectors. One may want, e.g., to minimize the functional called the *distortion measure*:

$$E_{VQ} = \int \|\mathbf{x} - \mathbf{m}_c\|^r p(\mathbf{x}) d\mathbf{x} , \tag{23}$$

where r is some real-valued exponent, the integral is taken over the complete metric $\mathbf{x}$ space, $\mathbf{m}_c$ is the $\mathbf{m}_i$ closest to $\mathbf{x}$, i.e.,

$$c = \arg\min_i \{\|\mathbf{x} - \mathbf{m}_i\|\} , \tag{24}$$

the norm is usually assumed Euclidean, $p(\mathbf{x})$ is the probability density function of $\mathbf{x}$, and $d\mathbf{x}$ is a shorthand notation for the n-dimensional volume differential of the integration space. All the values of $\mathbf{x}$ that have the same $\mathbf{m}_c$ as their nearest neighbor are said to constitute the *Voronoi set* associated with $\mathbf{m}_c$. Under rather general conditions one can determine the point density $q(\mathbf{x})$ of the $\mathbf{m}_i$ as in the following expression [2, 8]:

$$q(\mathbf{x}) = \text{const.} \left[p(\mathbf{x})^{\frac{n}{n+r}}\right] . \tag{25}$$

A related problem occurs with the *self-organizing map (SOM)*, which resembles VQ, but in which the $\mathbf{m}_i$ are *ordered* in $\mathbb{R}^n$ according to their similarity. The SOM carries out a vector quantization, too, but the placement of the $\mathbf{m}_i$ in the signal space is restricted by the neighborhood relations.

A long-standing problem has been whether the SOM model vectors could be determined by the minimization of some objective function. For instance, Kohonen, 1991 [3] discussed the distortion measure

$$E = \int \sum_i h_{ci} \|\mathbf{x} - \mathbf{m}_i\|^2 p(\mathbf{x}) d\mathbf{x} = \sum_i \int_{\mathbf{x} \in V_i} \sum_j h_{ij} \|\mathbf{x} - \mathbf{m}_j\|^2 p(\mathbf{x}) d\mathbf{x} . \tag{26}$$

where V_i is the Voronoi set around $\mathbf{m}_i$. The gradient of E consists of two terms :

$$\frac{\partial E}{\partial \mathbf{m}_j} = G + H , \tag{27}$$

where G is obtained if the integration borders are kept fixed and the differentiation with respect to $\mathbf{m}_j$ is carried out in the integrand only, whereas in the computation of H, the integrand is held constant and the integration borders are let to vary when the $\mathbf{m}_j$ differential is taken.

In order to avoid the evaluation of the above integrals, one may try to resort to the classical method called the *stochastic approximation* [7]. If the inputs $\mathbf{x}$ are obtained as a sequence of samples $\{\mathbf{x}(t)\}$, one can compute at every time t the best tentative estimate of $\mathbf{m}_i$ so far, called $\mathbf{m}_i(t)$. The expression

$$E_1(t) = \sum_i h_{ci} \|\mathbf{x}(t) - \mathbf{m}_i(t)\|^2 \tag{28}$$

is taken as the sample of function E at time t. Following Robbins and Monro, at time t we approximate the gradient of E with respect to $\mathbf{m}_i$ by the gradient of $E_1(t)$ with respect to $\mathbf{m}_i(t)$. Then

$$\mathbf{m}_i(t+1) \;=\; \mathbf{m}_i(t) - \left(\frac{\varepsilon}{2}\right) \frac{\partial E_1(t)}{\partial \mathbf{m}_i(t)} \tag{29}$$

with ε a small number. However, it is not yet clear how good an approximation the Robbins-Monro process is in this case. We have now shown that the *point density derived from the SOM algorithm* and the *point density derived from the SOM distortion measure* are different already in the one-dimensional case.

3.2 Point Densities in a Simple One-Dimensional SOM

3.2.1 Asymptotic State of the One-Dimensional Finite-Grid SOM Algorithm

Consider a series of samples of the input $x(t) \in \mathbb{R}$, $t = 0, 1, 2, \ldots$ and a set of k model (codebook) values $m_i(t) \in \mathbb{R}$, $t = 0, 1, 2, \ldots,$ whereupon i is the model index ($i = 1, \ldots, k$). For convenience assume $0 \leq x(t) \leq 1$.
The original one-dimensional self-organizing map (SOM) algorithm with at most one neighbor on each side of the best-matching m_i reads (Kohonen, 1997):

$$\begin{aligned}
m_i(t+1) &= m_i(t) + \varepsilon(t)[x(t) - m_i(t)] \;\text{ for }\; i \in N_c\,, \\
m_i(t+1) &= m_i(t) \;\text{ for }\; i \notin N_c\,, \\
c &= \arg\min_i\{|x(t) - m_i(t)|\}\,, \text{ and} \\
N_c &= \{\max(1, c-1), c, \min(k, c+1)\}\,,
\end{aligned} \tag{30}$$

where N_c is the neighborhood set around node c, and $\varepsilon(t)$ is a small scalar value called the learning-rate factor. In order to analyze the asymptotic values of the m_i, let us assume that the m_i are already ordered. The Voronoi set V_i around m_i is

$$\begin{aligned}
\text{for } 1 < i < k,\; V_i &= \left[\frac{m_{i-1} + m_i}{2}, \frac{m_i + m_{i+1}}{2}\right]\,, \\
V_1 &= \left[0, \frac{m_1 + m_2}{2}\right],\; V_k = \left[\frac{m_{k-1} + m_k}{2}, 1\right]\,, \;\text{ and denote} \\
\text{for } 1 < i < k,\; U_i &= V_{i-1} \cup V_i \cup V_{i+1}\,, U_1 = V_1 \cup V_2,\; U_k = V_{k-1} \cup V_k\,.
\end{aligned} \tag{31}$$

One can write the condition for stationary equilibrium of the m_i for a constant ε as:

$$\forall i,\;\; \mathrm{E}\,\{x - m_i | x \in U_i\} = 0\,. \tag{32}$$

For $2 < i < k-1$ we have for the limits of the U_i:

$$A_i = \frac{1}{2}(m_{i-2} + m_{i-1}) \quad , B_i = \frac{1}{2}(m_{i+1} + m_{i+2}) \ . \tag{33}$$

For $i = 1$ and $i = 2$ we must take B_i as above, but $A_i = 0$; and for $i = k-1$ and $i = k$ we have A_i as above and $B_i = 1$.

Numerical example. Let $p(x) = 2x$ for $0 \le x \le 1$ and $p(x) = 0$ otherwise.
The stationary values of the m_i are defined by the set of nonlinear equations

$$\forall i, \ m_i = \mathrm{E}\{x|x \in U_i\} = \frac{2(B_i^3 - A_i^3)}{3(B_i^2 - A_i^2)} \tag{34}$$

and the solution of (34) is sought by the so-called *contractive mapping*. Let us denote $\mathbf{z} = [m_1, m_2, \ldots, m_k]^{\mathrm{T}}$. Then the equation to be solved is of the form $\mathbf{z} = f(\mathbf{z})$. Starting with the first approximation for $\mathbf{z}$ denoted $\mathbf{z}^{(0)}$, each improved approximation for the root is obtained recursively:

$$\mathbf{z}^{(s+1)} = f(\mathbf{z}^{(s)}) \ . \tag{35}$$

In the present case one may select for the first approximation of the m_i, e.g., equidistant values.
It may now be expedient to define the point density q_i around m_i as the inverse of the length of the Voronoi set, or $q_i = [(m_{i+1} - m_{i-1})/2]^{-1}$.
The problem expressed in a number of previous works, e.g., Ritter and Schulten (1986), Ritter (1991), and Dersch and Tavan (1995), is to find out whether q_i could be approximated by the functional form const.$[p(m_i)]^{\alpha}$. Previously this was only shown for the continuum limit, i.e. for an infinite number of grid points. The present numerical analysis allows us to derive results for finite-length grids, too. Assuming tentatively that the power law holds for the models m_i through m_j (leaving aside models near to the ends of the grid), we shall then have

$$\alpha = \frac{\log(m_{i+1} - m_{i-1}) - \log(m_{j+1} - m_{j-1})}{\log[p(m_j)] - \log[p(m_i)]} \ . \tag{36}$$

In Table 1, using $i = 4$ and $j = k-3$, between which the border effects may be assumed as negligible, the exponent α has been estimated for 10, 25, 50, and 100 grid points, respectively.

3.2.2 Optimum of the One-Dimensional SOM Distortion Measure with Finite Grid

In the previous example, (26) becomes

$$\begin{aligned} E &= 2\sum_i \sum_{j \in N_i} \int_{C_i}^{D_i} (x - m_j)^2 x dx \\ &= \sum_i \sum_{j \in N_i} m_j^2 (D_i^2 - C_i^2) - \frac{4}{3} m_j (D_i^3 - C_i^3) + \frac{1}{2}(D_i^4 - C_i^4) \end{aligned} \tag{37}$$

where the *neighborhood set of indices* N_i was defined in (30), and the borders C_i and D_i of the Voronoi set V_i are $C_1 = 0$, $D_k = 1$,

$$C_i = \frac{m_{i-1} + m_i}{2} \quad \text{for } 2 \leq i \leq k \text{ , and } D_i = \frac{m_i + m_{i+1}}{2} \quad \text{for } 1 \leq i \leq k-1 \text{ .} \tag{38}$$

The optimal values of the m_i are determined by the gradient method:

$$\forall i \text{ ,} \quad m_i(t+1) = m_i(t) - \lambda(t) \cdot \partial E / \partial m_i |_t \text{ ,} \tag{39}$$

where $\lambda(t)$ is a suitable small scalar factor. With $\lambda(t) > .01$ (even with $\lambda(t) = 10$) and starting with very different initial values for the m_i, the process has converged robustly to a unique global minimum. After computation of the optimal values $\{m_i\}$, the exponent α of the tentative power law was computed from (36) of the previous section and presented in Table 1 for different lengths of the grid. Clearly the cases discussed in Secs. 2.1 and 2.2 are qualitatively different.

Table 1: Exponent α derived from the SOM algorithm and the SOM distortion measure, respectively

Grid points	SOM algorithm	SOM distortion measure
10	0.5831	0.3281
25	0.5976	0.3331
50	0.5987	0.3333
100	0.5991	0.3331

3.3 Derivation of the VQ Point Density by the Calculus of Variations

The technique that will be used to approximate point densities for higher-dimensional SOMs will first be applied to the simpler VQ problem. If $p(\mathbf{x})$ is smooth and the placement of the $\mathbf{m}_i$ in the signal space is reasonably regular, one may try to approximate the Voronoi sets, which are polytopes in the n-dimensional space, by n-dimensional hyperspheres centered at the $\mathbf{m}_i$. This, of course, is a rough approximation, but it was in fact used already in the classical VQ papers [2, 8], and no better treatments exist for the time being.

Denoting the radius of the hypersphere by R, its hypervolume has the expression kR^n, where k is a numerical factor. If $p(\mathbf{x})$ is approximately constant over the polytope, the *elementary integral of the distortion* $\|\mathbf{x} - \mathbf{m}_i\|^r = \rho^r$ *over the hypersphere* is

$$D = nk \int_0^R p(\mathbf{x}) \cdot \rho^r \cdot \rho^{n-1} d\rho = \frac{nk}{n+r} \cdot p(\mathbf{x}) \cdot R^{n+r} \text{ ;} \tag{40}$$

notice that if $v(\rho)$ is the volume of the n-dimensional hypersphere with radius ρ, then $dv(\rho)/d\rho = nk\rho^{n-1}$ is the "hypersurface area" of the hypersphere.

The point density $q(\mathbf{x})$ is defined as $1/kR^n$. What we aim at first is the approximate "distortion density" that we denote by $I[\mathbf{x}, q(\mathbf{x})]$, where $q(\mathbf{x})$ is the point density of the $\mathbf{m}_i$ at the value $\mathbf{x}$:

$$I[\mathbf{x}, q(\mathbf{x})] = \frac{D}{kR^n} = \frac{n}{n+r} \cdot p(\mathbf{x}) \cdot R^r = \frac{np(\mathbf{x})}{n+r}[kq(\mathbf{x})]^{-\frac{r}{n}} \ . \tag{41}$$

In the continuum limit, the total distortion measure is the integral of the "distortion density" over the complete signal space:

$$\int I[\mathbf{x}, q(\mathbf{x})]d\mathbf{x} = \int \frac{np(\mathbf{x})}{n+r}[kq(\mathbf{x})]^{-\frac{r}{n}} d\mathbf{x} \ . \tag{42}$$

This integral is minimized under the restrictive condition that the sum of all quantization vectors shall always equal N; in the continuum limit the condition reads

$$\int q(\mathbf{x})d\mathbf{x} = N \ . \tag{43}$$

In the classical *calculus of variations* one often has to optimize a functional which in the one-dimensional case with one independent variable x and one dependent variable $y = y(x)$ reads

$$\int_a^b I(x, y, y_x)dx \ ; \tag{44}$$

here $y_x = dy/dx$, and a and b are fixed integration limits. If a restrictive condition

$$\int_a^b I_1(x, y, y_x)dx = \text{const.} \tag{45}$$

has to hold, the generally known Euler variational equation reads, using the Lagrange multiplier λ and denoting $K = I - \lambda I_1$,

$$\frac{\partial K}{\partial y} - \frac{d}{dx}\frac{\partial K}{\partial y_x} = 0 \ . \tag{46}$$

In the present case x is vectorial, denoted by $\mathbf{x}$, $y = q(\mathbf{x})$, and I and I_1 do not depend on $\partial q/\partial \mathbf{x}$. In order to introduce fixed, finite integration limits one may assume that $p(\mathbf{x}) = 0$ outside some finite support. Now we can write

$$I = \frac{nk^{-\frac{r}{n}}}{n+r} \cdot p(\mathbf{x}) \cdot [q(\mathbf{x})]^{-\frac{r}{n}} \ , \ I_1 = q(\mathbf{x}) \ , \ K = I - \lambda I_1 \ , \tag{47}$$

$$\frac{\partial K}{\partial q(\mathbf{x})} = -\frac{rk^{-\frac{r}{n}}}{n+1} \cdot p(\mathbf{x}) \cdot [q(\mathbf{x})]^{-\frac{n+r}{n}} - \lambda = 0 \, . \tag{48}$$

At every location $\mathbf{x}$ there then holds

$$q(\mathbf{x}) = C \cdot [p(\mathbf{x})]^{\frac{n}{n+r}} \ , \tag{49}$$

where the constant C can be solved by substitution of $q(\mathbf{x})$ into (43). Clearly (49) is identical with (25). We have now obtained the same result that earlier ensued from very intricate signal and error-theoretic probabilistic considerations.

3.4 The SOM Point Density Derived from the Distortion Measure for Equal Vector and Grid Dimensionalities

It is possible to carry out the following analysis with a rather general symmetric h_{ij}, but for simplicity, without much loss of generality, we may assume, like in the basic SOM theory, $h_{ij} = 1$ within a certain radius, relating to the distances measured along the grid from the node j; outside this radius $h_{ij} = 0$. This is called the *neighborhood* around grid point $\mathbf{m}_j$.

In the signal space this then means that if $p(\mathbf{x})$ and the point density of the $\mathbf{m}_i$ are changing slowly, in the first approximation we can take $h_{ij} = 1$ up to a distance aR from $\mathbf{m}_j$, where R is the radius of the hypersphere that approximates the Voronoi set V_j, and a is a numerical constant; in other words, the neighborhood shall contain a constant number of grid points everywhere over the SOM (except at the borders of the SOM).

For the elementary integral of the distortion *over the neighborhood* up to radius aR, with the exponent $r = 2$, we then obtain according to (40):

$$D = \frac{nk}{n+2} \cdot p(\mathbf{x}) \cdot (aR)^{n+2} \, , \tag{50}$$

and relating the "distortion density" to the "volume" of V_j,

$$I[\mathbf{x}, q(\mathbf{x})] = \frac{D}{kR^n} = \frac{na^{n+2}}{n+2} \cdot p(\mathbf{x}) \cdot [kq(\mathbf{x})]^{-\frac{2}{n}} \, . \tag{51}$$

We then directly obtain in analogy with equations (41) through (48) and taking $r = 2$ the result

$$q(\mathbf{x}) = C'[p(\mathbf{x})]^{\frac{n}{n+2}} \tag{52}$$

with another constant C' computed from the normalization condition.

Notice that (52), however, does not yet tell anything about the exponent if the *SOM algorithm* is used to determine the $\mathbf{m}_i$.

References

[1] Dersch, D.R., Tavan, P. (1995). Asymptotic level density in topological feature maps. *IEEE Trans. Neural Networks* 6:230-236.

[2] Gersho, A. (1979) Asymptotically optimal block quantization. *IEEE Trans. Inf. Theory* 25:373-380.

[3] Kohonen, T. (1991) Self-organizing maps: optimization approaches. In Kohonen, T., Mäkisara, K., Simula, O., Kangas, J. (Eds.), *Artificial Neural Networks* (vol. 2, pp. 981-990). Amsterdam: Elsevier.

[4] Kohonen, T. (1999) Comparison of SOM point densities based on different criteria. *Neural Computation*, in press.

[5] Ritter, H. (1991) Asymptotic level density for a class of vector quantization processes. *IEEE Trans. Neural Networks* 2:173-175.

[6] Ritter, H., Schulten, K. (1986) On the stationary state of Kohonen's self-organizing sensory mapping. *Biol. Cybern.* 54:99-106.

[7] Robbins, H., Monro, S. (1951) A stochastic approximation method. *Ann. Math. Statist.* 22:400-407.

[8] Zador, P.L. (1982) Asymptotic quantization error of continuous signals and the quantization dimension. *IEEE Trans. Inf. Theory* IT-28:139-149.

4 The Median SOM

Teuvo Kohonen

The model vectors $\mathbf{m}_i$ in the basic SOM were determined as conditional averages over selected subsets of samples $\mathbf{x}(t)$. Thereby, however, sharp structures in the patterns formed by the components of $\mathbf{x}(t)$ will be smoothed out. Also, if the SOM is used to represent sets of statistical descriptors of a discrete set of items (cf. Sec. 11), the models will no longer be exact replica of any descriptor sets.
An alternative way for the construction of the SOM is to use the Batch Map principle, but instead of updating the old models $\mathbf{m}_i(t)$ by the respective means over the unions of the Voronoi sets, one can take the so-called *set medians* over the unions for the updated values of the $\mathbf{m}_i(t)$.
The set median M over the set $S = \{X(t)\}$ is defined to be that member of S, the sum of distances of which from all the other elements of S is minimum:

$$\sum_t d[X(t), M] = \min ! \tag{53}$$

The reason for calling M the "median" is that if the $X(t)$ are scalar numbers, and if $d[X(t), M] = |X(t) - M|$, then M is easily seen to be the arithmetic median of S. However, in the most general case, the $X(t)$ need not even be vector-valued.

The Batch Map algorithm is thus modified in the following:

1. Initialize the models M_i in some proper way.
2. Input all the available samples $X(t)$, and list each of them under the respective winner unit c (for which $d[X(t), M_i]$ is minimum). If there are several winners for $X(t)$, select one of them at random for listing.
3. Take for the updated value of M_i the median over the neighborhood N_c, i.e., over the union of the above lists associated with the winner unit c and its neighborhood N_c. If there are several medians, select one of them at random for the effective median.
4. Repeat from 2 a few times, until the values of the M_i can be regarded as steady.

In order to speed up the computations, if the usual SOM or Batch Map algorithm is applicable to the $X(t)$, it may be advisable to first construct the SOM in the traditional way and after that continue using the median algorithm.
As the set median is a replica of some of the members, the internal structures of this member will be preserved in the mapping, and thus every model resulting in the SOM will always represent some real input sample.

5 Self-Organizing Maps of Symbol Strings

Teuvo Kohonen and Panu Somervuo

The SOMs are usually defined in metric vector spaces. A different idea altogether is organization of *symbol strings* or other nonvectorial representations on a SOM array, whereupon the relative locations of the images of the strings on the SOM are expected to reflect some distance measure, e.g., the *Levenshtein distance* or *feature distance (FD)*, between the strings (for textbook accounts, cf. [1,2]). If one tries to apply the SOM algorithm to such entities, the difficulty immediately encountered is that *incremental learning laws cannot be expressed for symbol strings*, which are discrete entities. Neither can a string be regarded as a vector.

It has recently transpired [3] that the SOM philosophy is amenable to the construction of ordered similarity diagrams for string variables, too. This method applies the following idea, earlier partly reported in Sec. 4: The *Batch Map* principle [2] (cf. also Sec. 1) is used to define learning as recursively computed *set medians, generalized medians, set means*, or *generalized means* [4] over sets of strings.

An additional advantage, not possessed by the vector-space methods, is obtained if the feature distance measure for strings is applied. The best match between the input string against an arbitrary number of reference strings can then be found directly by the so-called *Redundant Hash Addressing (RHA)* method [2,5]. In it, the number of comparison operations, in the first approximation at least, *is almost independent of the number of model strings*. Construction of very large SOM arrays for strings becomes then possible.

Let us recall that the *set median* M over $\mathcal{S} = \{X(t)\}$ was defined by

$$\sum_t d[X(t), M] = \min! \qquad (54)$$

where $d[X(i), X(j)]$ is the general distance between elements $X(i), X(j) \in \mathcal{S}$, and $M \in \mathcal{S}$. Similarly, the *set mean* m over $\mathcal{S}$ shall satisfy the condition

$$\sum_t d^2[X(t), m] = \min! \qquad (55)$$

The *generalized median* and the *generalized mean* are then defined to result from the conditions (54) and (55) when M and m are not restricted to belong to $\mathcal{S}$. If $X(t) \in R^n$, and if m need not belong to $\mathcal{S}$, m is simply the arithmetic mean of the $X(t)$.

The basic types of error that may occur in strings of discrete symbols are: (1) replacement, (2) insertion, (3) deletion of a symbol. (Interchange of two consecutive symbols can be reduced to two of these operations.) An insertion or deletion error changes the relative position of all symbols to the right of it, whereupon, e.g., the most trivial distance between strings of symbols, the Hamming distance is not applicable. There are at least two categories of distance measures that take into account the "warping" of strings: (1) *Levenshtein distance*, which usually computes the minimum number of editing operations (replacements, insertions, and deletions of symbols) needed to change one string into another; these operations can also be weighted in many ways; (2) comparison of strings by their *local features*, e.g., substrings of N consecutive symbols (N-grams), whereupon the respective local features are said to match only if their relative position in the two strings differs in no more than a prespecified number of positions. The string lengths can also be taken into account [1, 2].

The set median and the set mean for strings are found easily, by computing all the mutual distances between the given strings, and searching for the string that has the minimum sum of the distances, or the minimum sum of squares of the distances, respectively, from the other strings. The generalized median and the generalized mean are then found by systematically varying each of the symbol positions of the set median or the set mean, making 'errors' of all the three types over the whole alphabet, and checking whether the sum of the distances or the sum of squares of the distances from the other elements is decreased. The computing time is usually quite modest; even with the 50 per cent error rate discussed here, the generalized median and the generalized mean can be found in the immediate vicinity of the set median and the set mean, respectively, in one or a couple of cycles of variation.

A number of additional problems has to be solved, too. One of them is *initialization* of the SOM with proper strings.

It is possible to initialize a usual vector-space SOM by random vectorial values. We have also been able to obtain organized SOMs for string variables, starting with random reference strings. However, it is of a great advantage if the initial values are already ordered, even roughly, along with the SOM array.

Ordered, although not yet optimal initial values of the strings can be picked up from the *Sammon projection* [2,6] of a sufficient number of representative input samples.

Another partial problem is *interpolation* between strings, especially if the dimensions of the SOM are changed during learning, as made in this work.

The most advantageous learning strategy for this method is to start with a very small SOM, and after its preliminary convergence, to halve the grid spacings intermittently by introducing new nodes in the middle of the old ones. If we input all the available samples to the smaller SOM and construct the partial lists at the matching nodes, then for the intermediate value to be used for the initialization of each middle node, we can take the average (median or mean) over the union of the lists collected for the neighboring nodes. After the first "expansion" and initialization of the middle nodes, the larger SOM is again taught by the available samples and "expanded," the new middle nodes are initialized in the same way, and so on, until the wanted size of the SOM is achieved.

SOMs of strings have been made for phonemic transcriptions produced by the speech recognition system similar to that reported in [7]. As feature vectors we used concatenations of three 10-dimensional mel-cepstrum vectors computed at successive intervals of time 50 ms in length. The phoneme-recognition and phoneme-decoding part was first tuned by the speech of nine male speakers using a 350-word vocabulary, after which the parameters of the system were fixed. The phoneme strings used in the following experiment were then collected from 20 speakers (15 male speakers and five female speakers). The string classes represented 22 Finnish command words. Finnish is pronounced almost like Latin. The results are shown in Fig. 2.

The classification accuracy of a usual SOM can be improved by supervised learning, fine tuning the reference vectors by the *Learning Vector Quantization (LVQ)* (cf.,

Table 2: Medians and means of garbled strings. *LD*: Levenshtein distance; *FD*: feature distance

Correct string: **MEAN**
Garbled versions (50 per cent errors):

1. MAN	6. EN
2. QPAPK	7. MEHTAN
3. TMEAN	8. MEAN
4. MFBJN	9. ZUAN
5. EOMAN	10. MEAN

Set median *(LD)*:	**MEAN**	**Set mean *(LD)*:**	**MEAN**
Generalized median *(LD)*:	**MEAN**	**Generalized mean *(LD)*:**	**MEAN**
Set median *(FD)*:	**MEAN**	**Set mean *(FD)*:**	**MEAN**
Generalized median *(FD)*:	**MEAN**	**Generalized mean *(FD)*:**	**MEAN**

Correct string: **HELSINKI**
Garbled versions (50 per cent errors):

1. HLSQPKPK	6. HOELSVVKIG
2. THELSIFBJI	7. HELSSINI
3. EOMLSNI	8. DHELSIRIWKJII
4. HEHTLSINKI	9. QHSELINI
5. ZULSINKI	10. EVSDNFCKVM

Set median *(LD)*:	HELSSINI	Set mean *(LD)*:	HELSSINI
Generalized median *(LD)*:	**HELSINKI**	**Generalized mean *(LD)***:	**HELSINKI**
Set median *(FD)*:	HELSSINI	Set mean *(FD)*:	HELSSINI
Generalized median *(FD)*:	HELSSINI	Generalized mean *(FD)*:	HELSSINI

e.g., [2]). It can be shown that a particular kind of LVQ is able to fine tune strings, too.
In accordance with the Batch-LVQ1 procedure introduced in Ref. [3] and also expounded in the first article of this report, we obtain the Batch-LVQ1 for strings by application of the following computational steps:

1. For the initial reference strings take, for instance, those strings obtained in the preceding SOM process.

2. Input the classified sample strings once again, listing the strings as well as their class labels under the winner nodes.

3. Determine the labels of the nodes according to the majorities of the class labels in these lists.

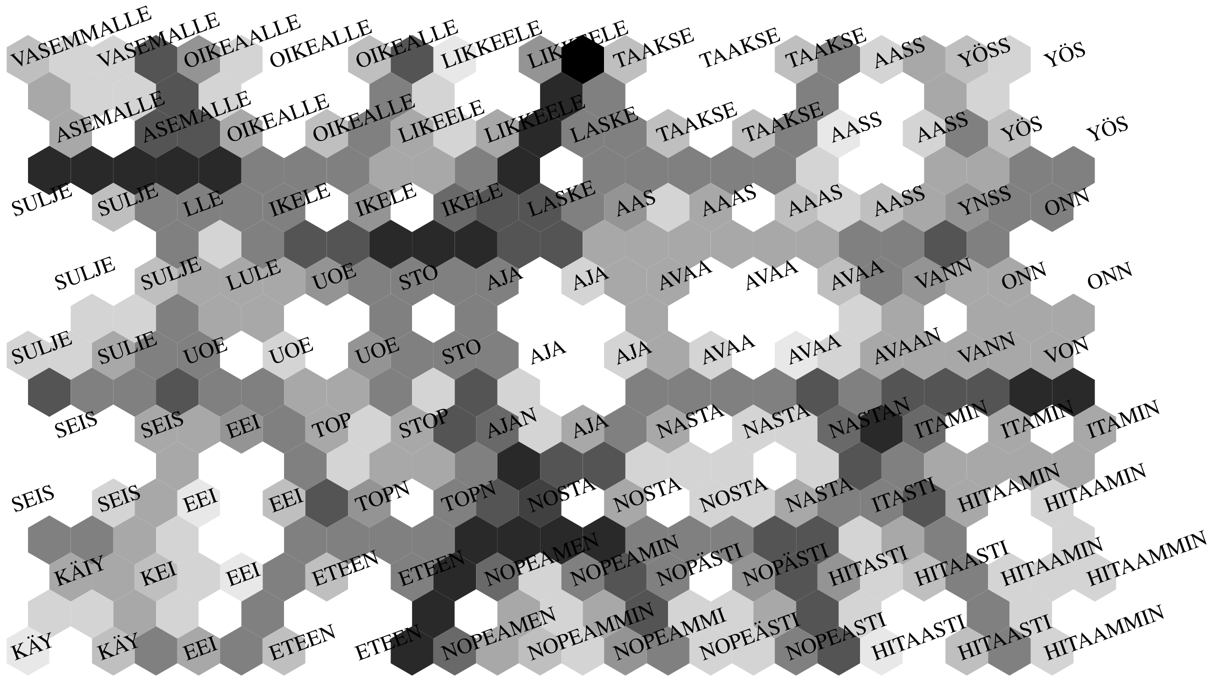


Figure 2: A 13 by 9 unit string-mean SOM. The shades of gray represent distances between neighboring reference vectors; dark means large distance, white small distance, respectively.

4. For each string in these lists, provide its distance (or its square of the distance) from every other string in the same list with the plus sign, if the class label of the latter sample string agrees with the label of the node, but with the minus sign if the labels disagree.

5. Take for the new value of the reference string the string that has the smallest sum of expressions defined at step 4 with respect to all the other strings in the respective list. Continue by systematically varying each of the symbol positions by replacement, insertion, and deletion of a symbol accepting the variation if the sum of expressions defined at step 4 is decreased. Take the best variation for the new reference string.

6. Repeat steps 1 through 5 a sufficient number of times.

The *multi-speaker word recognition* experiments for the 20 speakers were carried out using smaller (9 by 9) hexagonal SOM lattices than in the previous examples. After training of the SOMs, seven rounds of fine tuning by LVQ1 were performed. The training and test sets consisted of 880 words each. The recognition results are given in Table 3.

We can see from the experiments that the SOM alone may already yield a reasonably high recognition accuracy. For comparison, if the correct (linguistic) phonemic transcriptions had been used as reference strings, the error percentage would have remained higher: 5.8 per cent.

Table 3: Recognition experiments. Average error percentages of four independent runs

Median strings

	training set	test set
SOM only, generalized median	4.3	4.5
SOM only, set median	3.7	3.7
SOM + LVQ1	3.2	3.3

Mean strings

	training set	test set
SOM only, generalized mean	4.1	4.4
SOM only, set mean	4.0	4.0
SOM + LVQ1	2.6	2.7

References

[1] T. Kohonen. *Self-Organization and Associative Memory.* Springer Series in Information Sciences, vol. 8, Springer, Heidelberg, 1984.

[2] T. Kohonen. *Self-Organizing Maps*, Springer Series in Information Sciences, vol. 30, Springer, Heidelberg, 1995.

[3] T. Kohonen. Self-organizing maps of symbol strings. Report A42, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland, 1996.

[4] T. Kohonen. Median strings. *Patt. Rec. Lett.*, 3:309-313, 1985.

[5] T. Kohonen. *Content-Addressable Memories.* Springer Series in Information Sciences, vol. 1, Springer, Heidelberg, 1980.

[6] T. Kohonen, J. Hynninen, J. Kangas, and J. Laaksonen. SOM_PAK: The self-organizing map program package. Report A31, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland, 1996.

[7] K. Torkkola, J. Kangas, P. Utela, S. Kaski, M. Kokkonen, M. Kurimo, and T. Kohonen. Status report of the Finnish phonetic typewriter project. In *Artificial Neural Networks*, T. Kohonen et al. (eds.), Elsevier, Amsterdam, vol. 1, pp. 771-776, 1991.

6 The SOM as a Model of Brain Maps

Teuvo Kohonen

The original motivation for the SOM algorithm was an attempt to explain various spatially organized neural "maps" in the central nervous system. In the light of the present knowledge, however, it seems necessary to distinguish three categories of such "brain maps": A1. Feature-sensitive cells that respond, e.g., to specific sensory stimuli. A2. Anatomically organized representations of the body or some receptor surface of it, e.g., in the visual, somatosensory, motor, and auditory cortices. A3. Abstract feature maps that constitute a topographical or topological representation of a specific feature space of the sensory experiences. Examples of such abstract maps are the color map in the visual area V4, and various maps of the auditory space.

Unlike the anatomical maps, the ordered mappings of abstract features cannot be produced by genetically controlled ordered growth of axons, because no ordered receptor system, from which such ordered axons could originate, exists for abstract features. The order that has ensued in the mapping must have emerged by self-organization.

The following three conditions seem to be necessary for the production of biological maps of abstract features: B1. All the cells of such a brain area must receive essentially similar information. B2. There must exist a mechanism for the activation of that particular cell (called the "winner") which, in some sense, is "best fit" to the input information. Its activity shall further be enhanced, for instance by lateral excitation and inhibition, while the activity in the rest of cells is suppressed. B3. There must exist a learning mechanism by which the "winner" and a subset of its spatial neighbors in the area become "tuned" to the prevailing stimulus, while no learning outside this subset occurs. In the long run, when different subsets of cells are activated by different stimuli, a global order along with some dominant features in the stimuli then ensues.

It will be necessary to notice that the above conditions B1 and B2 entail that an area over which the input can be regarded similar, and in which lateral interactions over the area may occur, cannot be very large. In view of the extent to which the afferent axons and intracortical collaterals can spread, a feature map in the cortex may only have a diameter of a few millimeters.

In the biological modeling it is customary to approximate the activation of a neuron by the dot product of its input signal vector $\mathbf{x}$ and its synaptic weight vector $\mathbf{m}_i$, but if this law is used for the definition of the winner in the SOM, then the updating law must be made *self-normalizing*:

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + h_{ci}[\mathbf{x}(t) - (\mathbf{m}_i^{\mathrm{T}}(t)\mathbf{x}(t))\mathbf{m}_i(t)] \; , \tag{56}$$

where h_{ci} is the neighborhood function, and the winner is defined by

$$c = \arg\max_i \{\mathbf{m}_i^{\mathrm{T}}(t)\mathbf{x}(t)\} \; . \tag{57}$$

In the next report we shall show how the maximum selection in (57), in principle at least, could be implemented by physiologically plausible networks.

It can be shown that starting with arbitrary initial values $\mathbf{m}_i(0)$, with $\|\mathbf{x}(t)\| = 1$, and h_{ci} sufficiently small, the $\|\mathbf{m}_i(t)\|$ tend to the value 1. Nonetheless (56) preserves the self-organizing property, which can be seen, e.g., from Fig. 3, where the phoneme map has been computed using this equation.

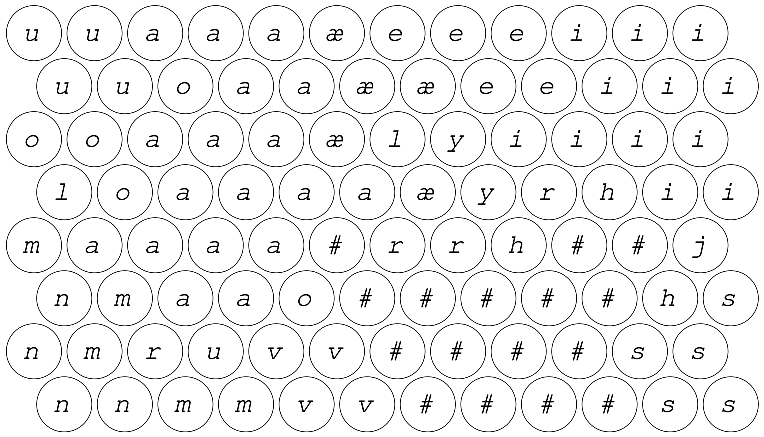


Figure 3: Self-organizing map of Finnish phonemes when 15-channel short-time spectra of natural speech, evaluated at every 20 ms, were used as the $\mathbf{x}(t)$ in (56).

The biological motivation for the bracketed expression in (56) may come from the following argumentation. First, the adaptive changes of a synapse must be made *reversible* in order to keep its weight on the dynamic range during its whole lifetime. Such a law, in the discrete-time formalism, could have the general form

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \alpha[P\mathbf{x}(t) - Q\mathbf{m}_i(t)] \,, \tag{58}$$

where α is a small factor. The term $\alpha P\mathbf{x}(t)$ describes the memory traces due to all presynaptic excitations, and the strengths of the memory traces are assumed proportional to $\mathbf{x}(t)$, while $-\alpha Q\mathbf{m}_i(t)$ represents the *forgetting effect*, which is assumed proportional to $\mathbf{m}_i(t)$. It may be stipulated that the biological forgetting is mostly "active," e.g., Q depends on the degree of activation of the cell, being proportional to $\mathbf{m}_i^{\mathrm{T}}(t)\mathbf{x}(t)$. Furthermore, in order to hold the "memory traces" steady for indefinite periods of time, while being able to change them fast upon demand, it must be assumed that both learning and forgetting (i.e., in general the synaptic changes) not only depend on the presynaptic signal activity, but are also conditioned by some *plasticity control factor* or *"learning factor"* produced by strong activities either at the cell itself or at its *neighboring cells*. If then the activity of the neural network is strongly clustered, i.e., if some kind of competitive process is at work selecting the winner, enhancing its activity, and suppressing the activity in the rest of the cells, then spatial spreading of the "learning factor" to neighboring cells means that P and Q must involve some *interaction kernel* h_{ci} as a factor, relating to the active cell c and cell i. This argumentation then directly results in the adaptation law (56).

Reference

T. Kohonen and R. Hari. Where the abstract feature maps of the brain might come from. *Trends in Neurosciences*, 22:135-139, March 1999.

7 Winner-Take-All (WTA) Network

Samuel Kaski and Teuvo Kohonen

In the practical SOM algorithms, selection of the winner by arithmetic computation is no problem. However, as the biological neural networks must implement this computation by dynamical components and networks with simple structures, special solutions compatible with the real neurophysiology must then be sought.
The winner index c in the biologically motivated SOM was defined by

$$c = \arg\max_i \{\mathbf{m}_i^{\mathrm{T}}\mathbf{x}\} \ . \tag{59}$$

In modeling, the dot products $\mathbf{m}_i^{\mathrm{T}}\mathbf{x}$ correspond to the total postsynaptic activations I_i of the neurons. They are formed directly at the inputs of the neurons. Therefore, it will remain necessary to study under what conditions a physiologically plausible simple network structure can select the largest of its scalar inputs (activations), i.e., implement the winner selection. Such a circuit is called the "winner-take-all" (WTA) network. For an early approach to this problem, cf. [1,2].
Our analysis is potentially applicable to any network in which the connections coming to each neuron can be grouped into external input, self-feedback, and feedback from the other neurons within the network (Figure 4). We used a neuron model introduced earlier [5], which describes changes in the activity, averaged spiking frequency η, of a cell as a function of the external inputs to the cell, I, and a nonlinear *convex* loss function γ:

$$d\eta/dt = I - \gamma(\eta) \ . \tag{60}$$

The nonlinear loss function represents the resultant of all losses and the effect of the refractory time of the cell. (To be exact, equation 60 holds only when $\eta > 0$ or when the right-hand side is positive, since spiking activity must always be positive.)
In the simplest network that we analyze, the input to neuron i consists of the external input coming from outside of the network, I_i^e, self-feedback from the neuron to itself, $g^+\sigma(\eta_i)$, and the feedback from the other cells, $g^- \sum_k \sigma(\eta_k)$. Here g^+ and g^- are coefficients that determine the strength of the connections, and σ models the combined effects of the transfer functions of the possible interneurons and any saturating nonlinearities on the signals. The dynamical system formed of the neural network can be described with the following set of differential equations:

$$d\eta_i/dt = I_i^e + g^+\sigma(\eta_i) + g^- \sum_k \sigma(\eta_k) - \gamma(\eta_i) \ , \tag{61}$$

$i = 1, \ldots, N$, where N is the number of neurons in the network.
This simple network type had already been analyzed previously [6], but now it turned out that the analysis could be generalized [4] to networks with several types of even nonidentical feedback connections (interneurons). To make the analysis most general the system of differential equations generalized from (61) was dressed mathematically into the form of a certain class of dynamical systems,

$$dy_i/dt = \lambda(y_i)[a_i(y_i) + b(y)] \ , \tag{62}$$

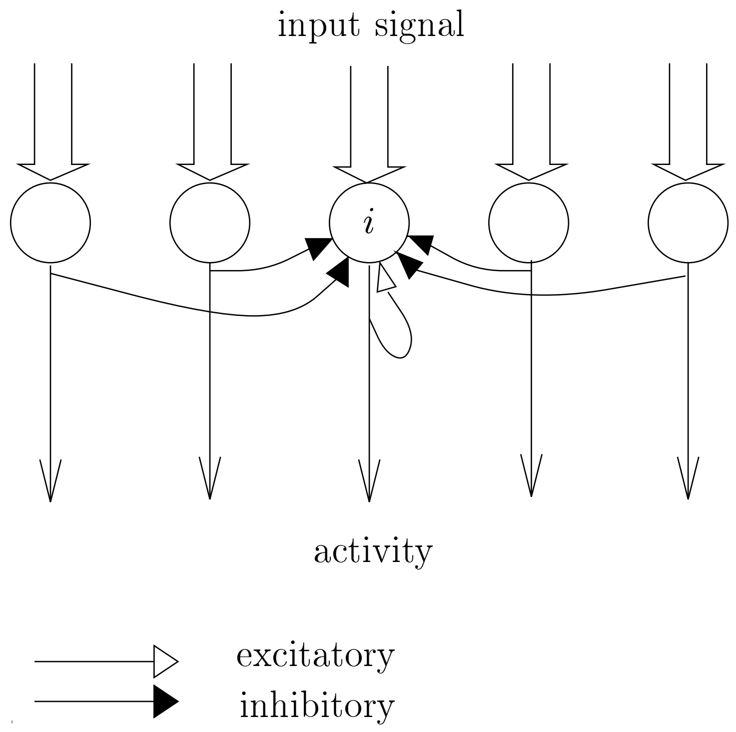


Figure 4: A schematic winner-take-all network. The neurons compete through the negative (inhibitory) feedback connections. The neuron receiving the largest input will be the only neuron that remains active after the initial transient activity. Only the connections coming to neuron i are shown.

where λ, a_i, and b are certain functions, and y is a vector formed of all the state variables y_i. Convergence properties of these types of systems had already been analyzed in [3].

It was then possible to prove that if certain restrictions are placed on the functions λ, a_i, and b, only one of the state variables y_i remains above a threshold, whereas the rest of them remain below a lower threshold. The lower threshold is zero for the neuron models. This is the essence of any WTA function.

When this more general analysis [4] was applied to the more general neural network models, conditions under which the networks have the WTA property were obtained. The most important conditions concern the external input: one of the neurons must receive the largest input, and all the other inputs must lie within a sensible range that does not depend on the largest input; otherwise also some other neurons may become active. In the beginning of the competition the winner must also be at least as active as the other neurons, which is the case, e.g., in the more complete network model described later in this section. The other, very mild conditions concern the form and steepness of the loss-function γ and the conductance function σ, and the strengths of the feedback connections g^+ and g^-.

The essential novelty in our analyses was their generality. Nevertheless, the models incorporated the common assumption that "sigmoidal"-type nonlinear transfer functions (function σ in 61) are adequate for modeling the effects of interneurons. It was possible, however, to further generalize the analyses by modeling the interneurons explicitly; the system of differential equations (61) then includes one extra equaa

b

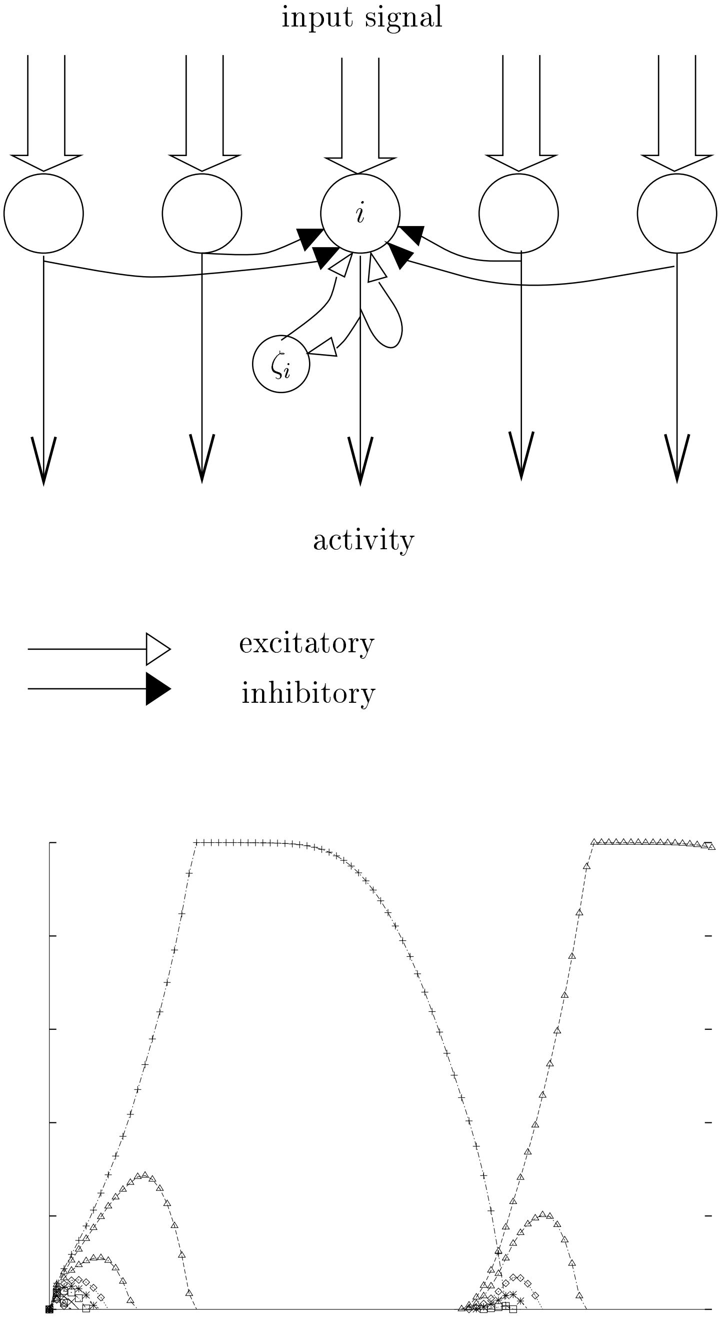


Figure 5: **a** Auxiliary slower inhibitory interneurons (marked with ζ_i in the schematic network) inactivate the active neuron after a brief interval, whereafter the competition may start again. If the inputs have changed meanwhile the previous winner was active, the new winner will be the one receiving the largest input. Otherwise the "runner-up", the neuron receiving the second-largest input will win. **b** A sample period of the activities of the neurons in a 20-neuron network.

tion for each interneuron. It is not possible, however, to guarantee in general that such models converge, but we were able to give general conditions under which the convergent models are WTA networks.

The WTA networks in which the winner *remains* active after it has become active are of course not sufficient models of the activity in physical neural networks. We coined such networks *weak* WTA circuits. In practice a network must be able to follow the changing activity it receives — we called networks in which a new unit becomes active when the inputs change *strong* WTA networks. We have demonstrated that the networks we studied are strong WTA networks if there are certain auxiliary slower interneurons in the network. These neurons provide negative feedback that in effect resets the activity of the winning neuron after a certain period of time (Figure 5).

We may summarize our analyses by concluding that the network structure of the type schematized in Figure 4 has been shown to be very robust in implementing the competition that is a necessary precursor of the Self-Organizing Map, or in fact any competitive learning application.

References

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

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

[3] S. Grossberg. Contour enhancement, short term memory, and constancies in reverberating neural networks. *Studies in Applied Mathematics*, L11:213–257, 1973.

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

[5] T. Kohonen. An introduction to neural computing. *Neural Networks*, 1:3–16, 1988.

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

8 The Adaptive-Subspace Self-Organizing Map (ASSOM)

Teuvo Kohonen, Samuel Kaski, and Harri Lappalainen

A long-standing goal in our research has been to find out how certain invariant-feature filters may emerge in learning processes. This problem was recently solved by one of the authors [1-3]. The key insight was that if input patterns must be recognizable invariantly to certain transformations, the members in *natural sequences of* such patterns must also be produced from each other by the same transformations. If the sequences are relative short, one may think that a particular transformation predominates in them, and the successive patterns then belong to some linear subspace that corresponds to this transformations. Such signal subspaces can be learned by the architecture delineated in Fig. 6.

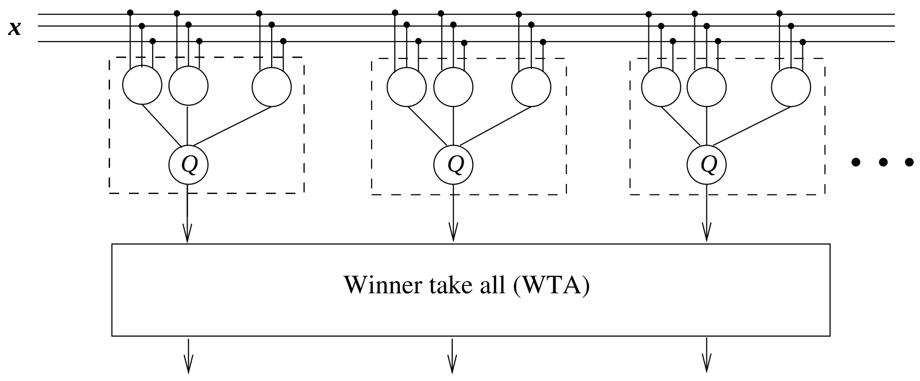


Figure 6: The ASSOM architecture.

Each dotted line in Fig. 6 distinguishes a module, a processing unit in a special SOM array. The first-layer neurons are linear and they output the sums of dot products of $\mathbf{x}$ with the various synaptic input weight vectors. The second-layer neurons (Q) form quadratic functions of the first-layer neuron outputs. If the weight vectors of the linear layer are orthonormalized, the neurons of the output layer shall form sums of squares of their inputs. The circuit represented by Fig. 6 can then be shown to compare the input pattern $\mathbf{x}$ with the linear subspaces spanned by the weight vectors of the first layer. If the weight vectors can be defined in a way in which their linear combinations represent some transformation groups, then matching becomes invariant with respect to these groups. Below it will be shown that such weight vectors emerge in an unsupervised learning process.

The outputs from the modules shall further be compared by a winner-take-all (WTA) function, which in Fig. 6 is shown as a separate operation. The WTA function specifies the "winner" module, indexed by c, as defined below; module c and its neighboring modules in the array will be updated in proportion to the so-called neighborhood function h_{ci} like in a usual SOM. In a physical network the WTA function may be integrated with the modules, for instance by their lateral interaction.

Let us denote the input weight vector, indexed by h of module i by $\mathbf{b}_h^{(i)}$. The $\mathbf{b}_h^{(i)}$ of the same module are now assumed orthonormal; at least they can be orthonor-

malized easily. They can then be regarded as the *orthonormal basis vectors of some linear subspace* $\mathcal{L}^{(i)}$, or a set of vectors $\mathbf{x}$, where every x can be expressed as a general linear combination of the $\mathbf{b}_h^{(i)}$.
Let $\hat{\mathbf{x}}^{(i)}$ be the *orthogonal projection* of $\mathbf{x}$ on $\mathcal{L}^{(i)}$, or

$$\hat{\mathbf{x}}^{(i)} = \sum_h \mathbf{b}_h^{(i)^{\mathrm{T}}} \mathbf{x} \ . \tag{63}$$

For an arbitrary $\mathbf{x}$ that need not belong to $\mathcal{L}^{(i)}$ one can define its *distance* d from $\mathcal{L}^{(i)}$, defined by

$$d^2 = d^2(\mathbf{x}, \mathcal{L}^{(i)}) = \|\mathbf{x}\|^2 - \|\hat{\mathbf{x}}^{(i)}\|^2 \ . \tag{64}$$

The *vectorial projection error* is the residual

$$\tilde{\mathbf{x}}^{(i)} = x - \hat{\mathbf{x}}^{(i)} \ . \tag{65}$$

For an arbitrary $\mathbf{x}$, its *minimum projection error* can be defined as the distance of $\mathbf{x}$ *from the closest subspace* $\mathcal{L}^{(i)}$, and the "winner subspace" with index c is defined by

$$\|\tilde{\mathbf{x}}^{(c)}\| = \min_i\{\|\tilde{\mathbf{x}}^{(i)}\|\} \ , \text{ or} \tag{66}$$

$$\|\hat{\mathbf{x}}^{(c)}\| = \max_i\{\|\hat{\mathbf{x}}^{(i)}\|\} \ . \tag{67}$$

Our goal is to let all the modules of Fig. 6 approximate $\mathbf{x}$ by its different projections, and always select the module that produces the best approximation over the array. The objective function that defines the *average expected spatially weighted normalized squared projection error* is

$$E_1 = \int \sum_i h_{ci} \frac{\|\tilde{\mathbf{x}}^{(i)}\|^2}{\|\mathbf{x}\|^2} p(\mathbf{x}) d\mathbf{x} \ , \tag{68}$$

where h_{ci} is the neighborhood function that defines the interaction of modules c and i like in a usual SOM, and c is the index of the winner subspace $\mathcal{L}^{(i)} = \mathcal{L}^{(c)}$. Notice that c is a function of $\mathbf{x}$ and all the basis vectors $\mathbf{b}_h^{(i)}$.
Minimization of (68), i.e., selection of the basis vectors $\mathbf{b}_h^{(i)}$ for all subspaces $\mathcal{L}^{(i)}$ such that the *average expected distance of* $\mathbf{x}$ *from the closest subspace is minimized*, is a rather complicated process [1-3]. Some extra problems are caused by the stability of the recursion by which E_1 is minimized. Without quoting all the details it may be mentioned that if the Robbins-Monro stochastic approximation process [4] is used, the optimal values of the $\mathbf{b}_h^{(i)}$ are obtained in the recursion [5].

$$\mathbf{b}_h^{(i)}(t+1) = \mathbf{b}_h^{(i)}(t) + \lambda(t) h_{ci}(t) \frac{\mathbf{x}(t) x^{\mathrm{T}}(t)}{\|\mathbf{x}(t)\|^2} \mathbf{b}_h^{(i)}(t) \ . \tag{69}$$

Consider now an "episode" $\mathcal{S}$ that consist of a finite set of successive sampling times t_p; denote $\mathcal{S} = \{t_p\}$. The set of samples $X = \{\mathbf{x}(t_p) | t_p \in \mathcal{S}\}$ has to be recognized as one class, such that any member of X and even an arbitrary linear combination of the $\mathbf{x}(t_p), t_p \in \mathcal{S}$ shall be decoded by the same module of Fig. 6 (subspace $\mathcal{L}^{(i)}$). In learning, the vector set X, defined as the Cartesian product of the $\mathbf{x}(t_p), t_p \in \mathcal{S}$, must be taken as one batch, instead of optimizing the error using single patterns $\mathbf{x}(t_p)$ one at a time. The error minimization problem will now be modified by defining the new objective function in terms of the *average expected spatially weighted normalized squared projection error over the episodes*:

$$E_2 = \int \sum_{t_p \in \mathcal{S}} \sum_i h_{ci} \frac{\|\tilde{\mathbf{x}}^{(i)}(t_p)\|^2}{\|\mathbf{x}(t_p)\|^2} p(X) dX \ . \tag{70}$$

Here $p(X)$ is the joint probability density for the samples $\mathbf{x}(t_p), t_p \in \mathcal{S}$ that produce the Cartesian product set X, and dX is a shorthand notation meaning a volume differential in the Cartesian product space of the $\mathbf{x}(t_p)$.
Minimization of (70) defines the basis vectors $\mathbf{b}_h^{(i)}$ and a set of analyzers that are *optimally invariant to the transformations that occur in the input signal patterns.*
The Robbins-Monro stochastic approximation is applicable to the minimization of E_2, too, when the gradient step is made to consist of the whole episode $\mathcal{S}$. The learning phase is then desribed by the following equation:

$$\mathbf{b}_h^{(i)}(t+1) = \mathbf{b}_h^{(i)}(t) + \lambda(t) h_{cr}^{(i)} \sum_{t_p \in \mathcal{S}(t)} \frac{\mathbf{x}(t_p)\mathbf{x}^{\mathrm{T}}(t_p)}{\|\mathbf{x}(t_p)\|^2} \mathbf{b}_h^{(i)}(t) \ . \tag{71}$$

When $\lambda(t)$ is small, (71) is equivalent with the following learning process in which the basis vectors are formed by a product of elementary projection operators, each one corresponding to one pattern $\mathbf{x}(t_p), t_p \in \mathcal{S}$:

$$\mathbf{b}_h^{\prime(i)} = \prod_{t_p \in \mathcal{S}} \left[I + \alpha(t_p) h_{cr}^{(i)} \frac{\mathbf{x}(t_p)\mathbf{x}^{\mathrm{T}}(t_p)}{\|\hat{\mathbf{x}}^{(i)}(t_p)\| \, \|\mathbf{x}(t_p)\|} \right] b_h^{(i)}(t_p) \ . \tag{72}$$

The special learning-rate factor $\lambda = \alpha(t_p) \|\mathbf{x}(t_p)\| \,/ \|\hat{\mathbf{x}}^{(i)}(t_p)\|$ in (72) has been chosen for stability reasons.
There are several other minor details in the process that improve the algorithm [3,5]. We have produced various ASSOM filters for very different input data [1,2]. Here a simple demonstration, illustrating the basic idea, is shown.
Over the input field we generated patterns consisting of colored noise (white noise, low-pass filtered by a second-order Butterworth filter with cut-off frequency of 0.6 times the Nyquist frequency of the sampling lattice). The input episodes for learning were formed by taking samples from this data field. The mean of the samples was always subtracted from the pattern vector.
In the *translation-invariant filter* experiment, the episodes were formed by shifting the receptive field randomly into five nearby locations, the average shift thereby being ± 2 pixels in both dimensions. Fig. 8 shows the basis vectors $\mathbf{b}_{i1}$ and $\mathbf{b}_{i2}$, similar to Gabor filters, in a gray scale at each array point of a two-dimensional ASSOM. One should notice that the spatial frequencies of the basis vectors of the same unit are the same, but the $\mathbf{b}_{i1}$ and $\mathbf{b}_{i2}$ are mutually 90 degrees out of phase. (The absolute phase of $\mathbf{b}_{i1}$ can be zero or 180 degrees, though.)
The episodes for the *rotation filters* were formed by rotating the input field at random five times in the range of zero to 60 degrees, the rotation center coinciding with the

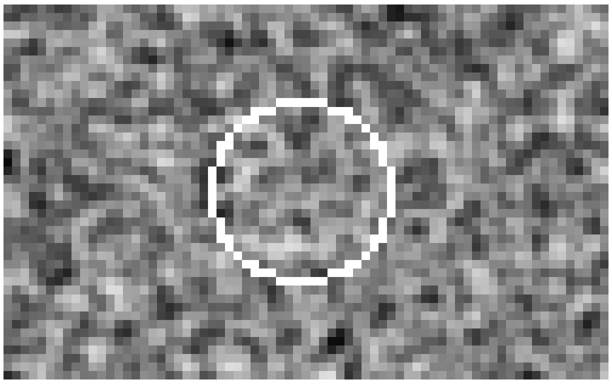
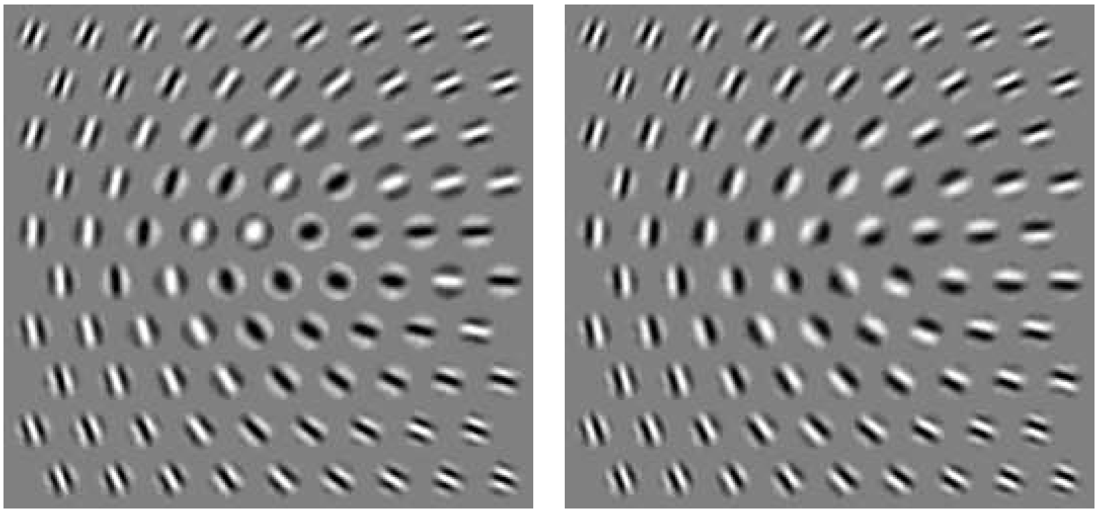


Figure 7: Colored noise (second-order Butterworth-filtered white noise with cut-off frequency of 0.6 times the Nyquist frequency of the lattice) used as input data. The receptive field is demarcated by the white circle.



(a) (b)

Figure 8: The ASSOM that has formed Gabor-type filters: (a) The $\mathbf{b}_{i1}$, (b) The $\mathbf{b}_{i2}$.

center of the receptive field. Fig. 9 shows the rotation filters thereby formed at the ASSOM units; clearly they are sensitive to azimuthal optic flow.

Scale-invariant filters were formed by zooming the input pattern field, with the center of the receptive field coinciding with the zooming center. The filters thereby formed, shown in Fig. 10, have clearly become sensitive to radial optic flow, corresponding to approaching or withdrawing objects.

References

[1] T. Kohonen. *Self-Organizing Maps.* Springer Series in Information Sciences, vol. 30. Springer, Heidelberg, 1995.

[2] T. Kohonen. Emergence of invariant-feature detectors in self-organization. In Palaniswami, M. et al. (eds.), *Computational Intelligence, A Dynamic System*

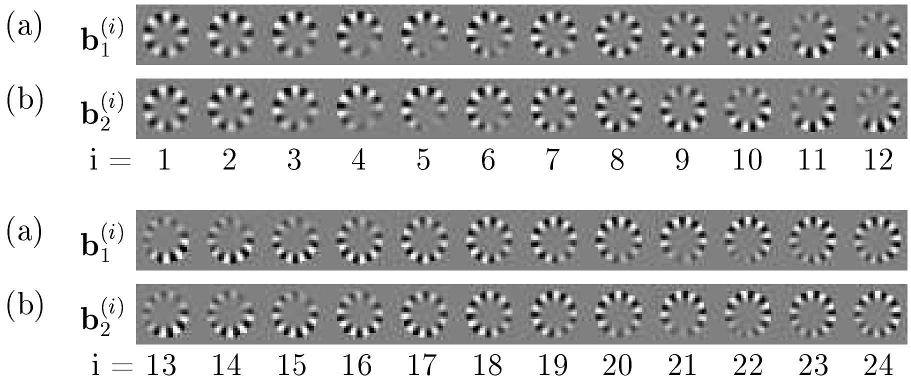


Figure 9: One-dimensional rotation-invariant ASSOM. (a) Cosine-type "azimuthal wavelets" ($\mathbf{b}_{i1}$), (b) Sine-type "azimuthal wavelets" ($\mathbf{b}_{i2}$). Notice that the linear array has been shown in two parts.

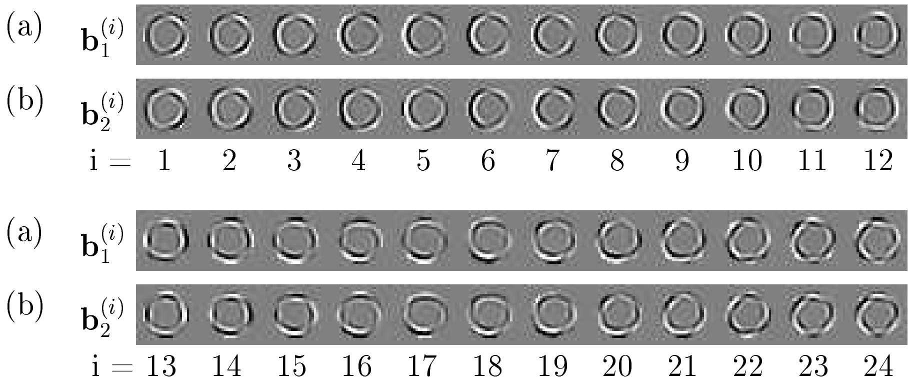


Figure 10: One-dimensional zoom-invariant ASSOM. (a) Cosine-type "radial wavelets" ($\mathbf{b}_{i1}$), (b) Sine-type "radial wavelets" ($\mathbf{b}_{i2}$). Notice that the linear array has been shown in two parts.

Perspective, pp. 17-31. IEEE Press, New York, 1995.

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

[4] H. Robbins and S. Monro. A stochastic approximation method. *Ann. Math. Statist.*, 22:400–407, 1951.

[5] T. Kohonen, S. Kaski, and H. Lappalainen. Self-organized formation of various invariant-feature filters in the adaptive-subspace SOM. *Neural Computation*, 9:1321-1344, 1997.

9 Speedup of SOM Computation

Teuvo Kohonen

9.1 Addressing Old Winners

If there are M map units (neurons) in the SOM, and for a certain statistical accuracy one stipulates that the number of updating operations per unit should be some constant (say, on the order of 100), then the total number of comparison operations to be performed by exhaustive search of the winners is $\sim M^2$.
Koikkalainen [1,2] has recently suggested a speedup method in which a search-tree structure, except its last layer, is replaced by pointers from data items to the next-to-last layer. A more accurate search is then made among the last branches of the tree. We will show below, however, that this idea is not restricted to tree structures, but can readily be added to any SOM software package. The total number of comparison operations can be made $\sim M$, provided that the training vectors have been given in the beginning, i.e., their set is finite and closed.
Assume that we are somewhere in the middle of the training process, whereby the last winner corresponding to each training vector has been determined; then the training vectors can be expressed as a linear table, with a *pointer* to the corresponding *tentative winner location* stored with each training vector (Fig. 11).

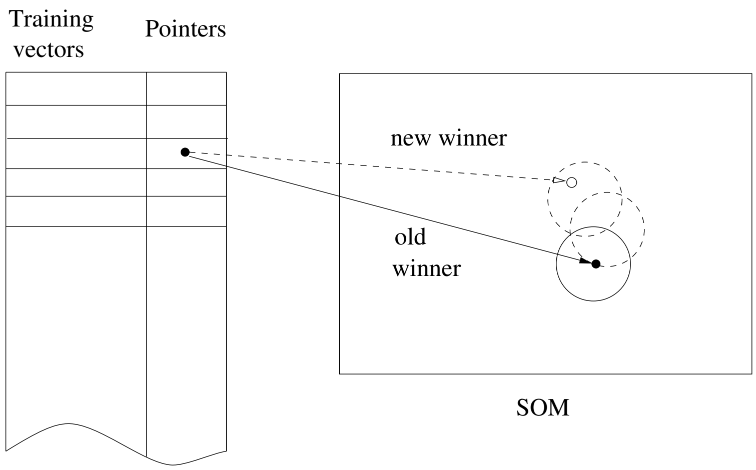


Figure 11: Finding the new winner in the vicinity of the old one, whereby the old winner is directly located by a pointer. The pointer is then updated

Assume further that the SOM is already smoothly ordered although not yet asymptotically stable. This is the situation, e.g., during the lengthy fine-tuning phase of the SOM, whereby the size of the neighborhood set is also constant and small. If, after inputting a particular input, updating of a number of map units is made before the same training input is used again some time later, it may be clear that the new winner is found at or in the vicinity of the old one. Therefore, in searching

for the best match, it will suffice to locate first the map unit corresponding to the associated pointer, and then to perform a local search for the winner in the neighborhood around the located unit. This will be a significantly faster operation than an exhaustive winner search over the whole SOM. The search can first be made in the immediate surround of the said location, and only if the best match is found at its edge, searching is continued in the surround of the preliminary best match, until the winner is one of the middle units in the search domain. After the new winner location has been identified, the associated pointer in the input table is replaced by the pointer to the new winner location.
For instance, if the array topology of the SOM is hexagonal, the first search might be made in the 7-neighborhood of the winner. If the tentative winner is one of the edge units of this neighborhood, the search must be continued in the new 7-neighborhood centered around the last tentative winner (for the three map units that have not yet been checked), etc.
This principle can be used with both the usual incremental-learning SOM and its batch computing version.
A benchmark with two large SOMs relating to our recent practical experiments was made. The approximate codebook vector values were first computed by the CNAPS computer, whereafter they were fine-tuned by a general-purpose computer. During this fine-tuning phase, the radius of the neighborhood set in the hexagonal lattice decreased linearly from 3 to 1 units equivalent to the smallest lattice spacings, and the learning-rate factor at the same time decreased linearly from 0.02 to zero. There were 3645 training vectors for the first map, and 9907 training vectors for the second map, respectively. The results are reported in Table 4.

Input dimensionality	Map size	Speedup factor in winner search	Speedup factor in training
270	315	43	14
315	768	93	16

Table 4: Speedup due to shortcut winner search.

The theoretical maximum of speedup in winner search is: 45 for the first map, and 110 for the second map, respectively. The training involves the winner searches, codebook updating, and overhead times due to the operating system and the SOM software used. The latter figures may be improved by optimization of computing.

9.2 Estimating Initial Values for a Large SOM

Several suggestions for "growing SOMs" (cf., e.g. [3-5]) have been made. The detailed idea presented below has been optimized in order to make very large maps, and is believed to be new. The basic idea is to estimate good initial values for a map that has plenty of units, on the basis of asymptotic values of a map with a much smaller number of units.
As the general nature of the SOM process and its asymptotic states is now fairly well known, we can utilize some "expert knowledge" here. One fact is that the asymptotic distribution of codebook vectors is generally smooth, at least for a continuous,

smooth probability density function (pdf) of input, and therefore the lattice spacings can be smoothed, interpolated, and extrapolated locally.
As an introductory example consider, for instance, the one-dimensional SOM and assume tentatively a uniform probability density function (pdf) of the scalar input in the range $[a, b]$. Then we have the theoretical asymptotic codebook values for different numbers of map units that approximate the same pdf, as shown in Fig. 12.

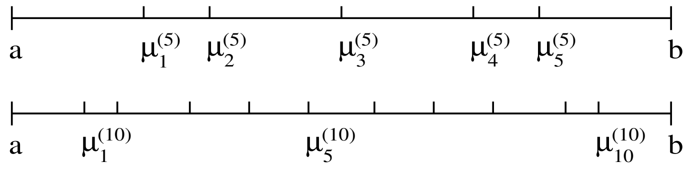


Figure 12: Asymptotic values for the μ_i for different lengths of the array, shown graphically

Assume now that we want to estimate the locations of the codebook values for an *arbitrary* pdf and for a 10-unit SOM on the basis of known codebook values of the 5-unit SOM. A linear *local* interpolation-extrapolation scheme can then be used. For instance, to interpolate $\mu_5^{(10)}$ on the basis of $\mu_2^{(5)}$ and $\mu_3^{(5)}$, we first need the *interpolation coefficient* λ_5, computed from the two ideal lattices with uniform pdf:

$$\mu_5^{(10)} = \lambda_5 \mu_2^{(5)} + (1 - \lambda_5)\mu_3^{(5)} , \tag{73}$$

from which λ_5 for $\mu_5^{(10)}$ can be solved. If then, for an arbitrary pdf, the *true* values of $\mu'^{(5)}_2$ and $\mu'^{(5)}_3$ have already been computed, the estimate of the true $\hat{\mu}'^{(10)}_5$ is

$$\hat{\mu}'^{(10)}_5 = \lambda_5 \mu'^{(5)}_2 + (1 - \lambda_5)\mu'^{(5)}_3 . \tag{74}$$

Notice that a similar equation can also be used for the *extrapolation* of, say, $\mu_1^{(10)}$ on the basis of $\mu_1^{(5)}$ and $\mu_2^{(5)}$.
Application of local interpolation and extrapolation to *two-dimensional* SOM lattices (rectangular, hexagonal, or other) is straightforward, although the expressions become a little more complicated. Interpolation and extrapolation of a codebook vector in a two-dimensional lattice must be made on the basis of vectors defined at least in *three* lattice points. As the maps in practice may be very nonlinear, the best estimation results are usually obtained with three reference vectors.
Consider a pdf that is uniform over a two-dimensional rectangular area, approximated by two different overlapping "ideal" two-dimensional SOM lattices with the codebook vectors $\mathbf{m}_h^{(d)} \in \Re^2, \mathbf{m}_i^{(s)} \in \Re^2, \mathbf{m}_j^{(s)} \in \Re^2$, and $\mathbf{m}_k^{(s)} \in \Re^2$ its nodes, where the superscript d refers to a "dense" lattice, and s to a "sparse" lattice, respectively. If $\mathbf{m}_i^{(s)}, \mathbf{m}_j^{(s)}$, and $\mathbf{m}_k^{(s)}$ do not lie on the same straight line, then in the two-dimensional signal plane any $\mathbf{m}_h^{(d)}$ can be expressed as the linear combination

$$\mathbf{m}_h^{(d)} = \alpha_h \mathbf{m}_i^{(s)} + \beta_h \mathbf{m}_j^{(s)} + (1 - \alpha_h - \beta_h)\mathbf{m}_k^{(s)} , \tag{75}$$

where α_h and β_h are interpolation-extrapolation coefficients. This is a two-dimensional vector equation from which the two unknowns α_h and β_h can be solved.

Consider then a pdf in a space of arbitrary dimensionality and two SOM lattices with the same topology as in the ideal example. When the true pdf is arbitrary, we may not assume the lattices of true codebook vectors as planar. Nonetheless we can perform a *local linear estimation* of the true codebook vectors $\mathbf{m}'^{(d)}_h \in \Re^n$ of the "dense" lattice on the basis of the true codebook vectors $\mathbf{m}'^{(s)}_i$, $\mathbf{m}'^{(s)}_j$, and $\mathbf{m}'^{(s)}_k \in \Re^n$ of the "sparse" lattice.
In practice, in order that the linear estimate be most accurate, we may stipulate that the respective indices h, i, j, and k are such that in the ideal lattice $\mathbf{m}^{(s)}_i, \mathbf{m}^{(s)}_j$, and $\mathbf{m}^{(s)}_k$ are the three codebook vectors *closest* to $\mathbf{m}^{(d)}_h$ in the signal space (but not on the same line). With α_h and β_h solved from (75) for each node h separately we obtain the wanted interpolation-extrapolation formula as

$$\hat{m}'^{(d)}_h = \alpha_h m'^{(s)}_i + \beta_h m'^{(s)}_j + (1 - \alpha_h - \beta_h) m'^{(s)}_k \; . \tag{76}$$

Notice that the indices h, i, j, and k refer to *topologically identical* lattice points in (75) and (76). The interpolation-extrapolation coefficients for two-dimensional lattices depend on their topology and the neighborhood function used in the last phase of learning. For the "sparse" and the "dense" lattice, respectively, we have to compute first the ideal two-dimensional codebook vector values. As the closed solutions may be very difficult to obtain, the asymptotic codebook vector values may be solved by simulation. If the ratio of the horizontal vs. vertical dimensions of the lattice is $H : V$, we may draw two-dimensional input vectors at random from a uniform, rectangular pdf, the width of which in the horizontal direction is H and the vertical width of which is V.

References

[1] P. Koikkalainen. Progress with the tree-structured self-organizing map. In *Proc. ECAI 94, 11th European Conf. on Artificial Intelligence*, A. Cohn (ed.), John Wiley & Sons, pp. 211-215, 1994.

[2] P. Koikkalainen. Fast determenistic self-organizing maps. In *Proc. ICANN, Int. Conf. on Artificial Neural Networks*, Paris, France, vol. 2, pp. 63-68, 1995.

[3] J.S. Rodrigues and L.B. Almeida. Improving the learning speed in topological maps of patterns. In *Proc. INNC'90, Int. Neural Networks Conference.* Kluwer, Dordrecht, Netherlands, pp. 813-816, 1990.

[4] B. Fritzke. Let it grow - self-organizing feature maps with problem dependent cell structure. In *Artificial Neural Networks*, T. Kohonen, K. Mäkisara, O. Simula, J. Kangas (eds.). North-Holland, Amsterdam, Netherlands, pp. I-403-408, 1991.

[5] J. Blackmore and R. Miikkulainen. Incremental grid growing: Encoding high-dimensional structure into a two-dimensional feature map. In *Proc. ICNN'93, Int. Conf. on Neural Networks.* IEEE, Piscataway, NJ, pp. I-450-455, 1993.

10 Fast Evolutionary Learning in the SOM

Teuvo Kohonen

Above we dealt with vector-valued $\mathbf{x}(t)$ and $\mathbf{m}_i(t)$. The SOM philosophy, however, can be much more general. The structures of the models and data can be different: it will suffice that some *fitness function* is definable between the general inputs X and the general models M_i, respectively. Let this function be denoted $f(X, M_i)$. *Notice carefully that we do not need any distance function in the X space, nor in the M_i space.*
Even under the above conditions, the SOM can be computed in an "evolutionary" process (cf. [1], Sec. 5.7). We may initialize the models as random samples from the set of possible models. Next we input the samples of X, one at a time, and at each step determine that model M_c for which

$$c = \arg\max_i \{f(X, M_i)\} \; . \tag{77}$$

The next step is some kind of *variation* of the M_i in the neighborhood set N_c of the fittest model M_c. This variation usually means random but statistically independent replacement of each M_i, $i \in N_c$ by some other possible model M on the condition $f(X, M) > f(X, M_i)$.
The evolutionary learning can be implemented by the batch-type SOM without random probing, whereupon it proceeds fast:

1. Initialize the models M_i, e.g., by a random choice of their parameter values from a set of possible values.

2. Input a number of items X and list each of them under the respective winner unit (i.e. that M_i for which some fitness function $f(X, M_i)$ is maximum). In case there is a tie, i.e., two or more M_i have the same fitness to X, select one of them randomly for the effective unique "winner" under which the listing is made.

3. Find a new value M_i' for each M_i such that if U_i is the union of lists relating to model M_i in the same way as in the Batch Map algorithm discussed in Sec. 1, the sum of the fitness-function values $f(X, M_i')$, $X \subset U_i$ is increased. If there exist ties, a random choice between the best M_i' is made.

4. Repeat from step 2.

References

[1] T. Kohonen, *Self-Organizing Maps*, Springer-Verlag, Berlin, 1995; 2nd edition, 1997.

[2] T. Kohonen, Fast evolutionary learning with batch-type self-organizing maps. To appear in *Neural Processing Letters*, 1999.

11 Statistical Data Analysis by the Self-Organizing Map

Samuel Kaski and Teuvo Kohonen

Knowledge discovery in databases (KDD) [1], sometimes also referred to as data mining, is a recently established field of research in which the aim is to discover novel patterns or structures in large data sets. The complex interactive discovery process involves several stages, of which the central stage called data mining refers to the application of essentially any suitable methods for finding interesting patterns in data.

KDD is related to a field of statistics called exploratory data analysis. Statistical inferences are often made in a two-stage process. Hypotheses are first generated in a data-driven phase, and the hypotheses are tested in another, confirmatory phase. The first methods for the exploratory, data driven phase were developed already in 1970's. The increase in computing power allows us to use much more sophisticated methods for looking at the statistical structures in data, and to analyze much larger data sets.

The central goal in exploratory data analysis is to present a data set in a form that is easily understandable but at the same time preserves as much essential information of the original data set as possible. The exploratory data analysis methods are general-purpose instruments that illustrate the essential features of a data set, like its clustering structure and the relations between its data items.

One may distinguish two categories of exploratory data analysis tools with somewhat different goals. First, some tools like the Sammon projection [5] *project* the multidimensional data set to, e.g., a two-dimensional plane while trying to preserve its whole structure (the distances between the data items) as well as possible. Other methods [3] try to find *clusters* in the data, whereby instead of the large data set only a small number of clusters needs to be considered.

A vast number of different algorithms to perform clustering is available. Choosing suitable algorithms and applying them correctly requires thorough knowledge of both the algorithms and the data set. There must exist enough clustering tendency in the data set in order that the use of clustering algorithms would be sensible at all, and as different clustering algorithms tend to find clusters of different shapes, the suitability of the shapes to describe the data set must be verified.

The projection methods, on the other hand, do not reduce the amount of data to be presented. Although they illustrate the essential features of the data set, the illustration is costly to obtain and may still be difficult to understand if the data set is large.

The self-organizing map algorithm is a unique method in that it combines the goals of both the projection and the clustering algorithms. It can be used at the same time to visualize the clusters in a data set, and to represent the set on a two-dimensional map in a manner that preserves the nonlinear relations of the data items; nearby items are located close to each other on the map. Moreover, even if no explicit clusters exist in the data set, the self-organizing mapping method reveals "ridges" and "ravines". The former are open zones with irregular shapes and high clustering tendency, whereas the latter separate data subsets that have a different statistical nature.

11.1 Case Study: Structures of Welfare and Poverty in the World

In this study we have demonstrated how the Self-Organizing Map is able to describe structures in a macroeconomic system. The map is shown to illustrate the "welfare or poverty states" of the countries of the world, when the data set describes different aspects of the standard of living. State transitions can easily be followed on the map. It is hoped that this study would serve as a recipe on how, using standard procedures, the state of any micro- or macroeconomic system can be presented in an easily understandable form. Only the data set needs to be changed in different applications.
The understanding and description of a complex entity like the standard of living requires simultaneous consideration of a large collection of statistical indicators describing its different aspects and their relationships. In this study we used a total of 39 indicators that described factors like health, education, consumption, and social services, picked up from the World Development Report of the year 1992 [7]. Based on the set of statistical indicators the SOM can be used to represent the welfare and poverty "states" of the countries on a *"poverty map"*.

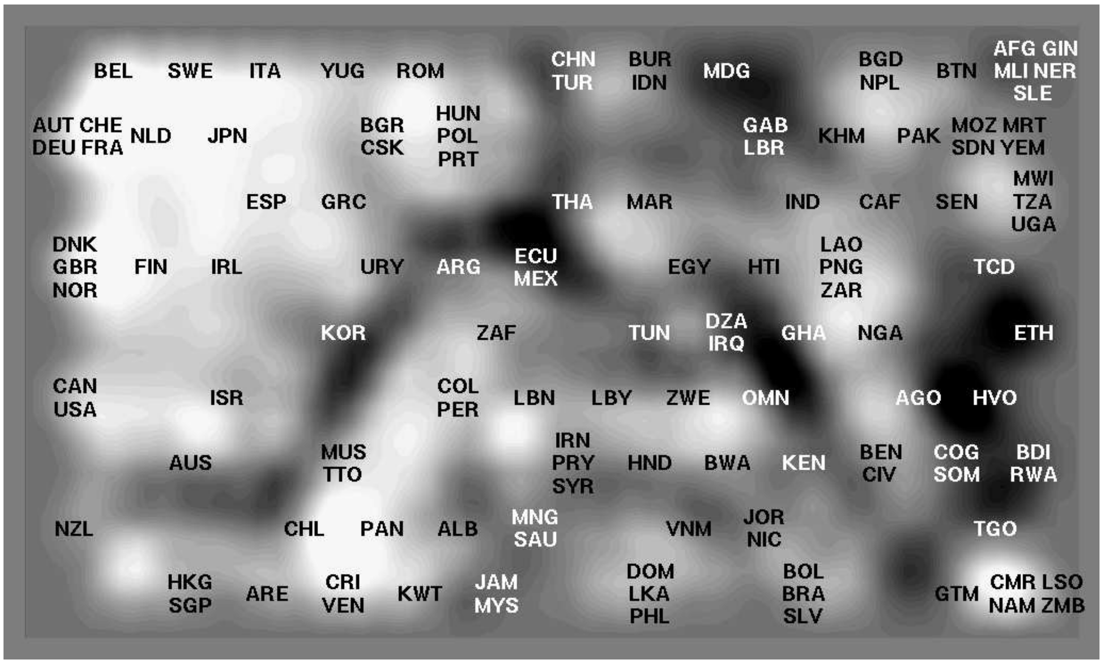


Figure 13: Structured diagram of the data set chosen to describe the standard of living. The order of the abbreviated country names indicates the similarity of the standard of living of the countries, and the colors indicate the degree of clustering. Light areas represent areas of a high degree of clustering and dark areas gaps in the degree of clustering. Different types of welfare and poverty are visible as the clustered (light) areas on the map, separated by the dark "ravines".

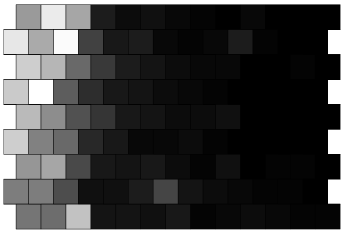


Figure 14: Distribution of the GNP per capita, which was not used in computing the maps, shown over the SOM groundwork. White indicates the largest value in the material and black the smallest, respectively. The horizontal axis of the map seems to correlate with the distribution of the overall welfare, as measured by the GNP per capita.

The clustering tendency in the data set can be visualized as a false-color or gray-scale display on the map (Figure 13; the display is a smoothed version of the so-called U-matrix [6] display; cf. also [2]). Different *types of welfare and poverty* are manifested on the display as clustered areas. For example, the cluster in the top left corner consists predominantly of the OECD countries, and an annex on the right of it is a cluster consisting mostly of the countries of Eastern Europe. It is evident from the display that most of the clustered areas are neither regularly shaped nor easily separated, but instead form some kinds of “hills”, “ridges” and “ravines”. It is then a definitive advantage of the method that no assumptions need to be made about the cluster shapes before the analysis, as is implicitly done in many other clustering methods.
The overall order of the countries on the map was found to illustrate the traditional conception of welfare; in fact, the horizontal dimension of the map seems to correlate fairly closely with the GNP (Gross National Product) per capita (Figure 14).
Refined interpretations about the fine structure of the welfare and poverty types, the clustered areas on the map display, can be made based on the original statistical indicators. The values of the indicators can be displayed in their natural order on the groundwork formed of the organized map. This display is much more easily understandable that ordinary linear statistical tables (examples have been shown in Fig. 15). Furthermore, such displays are readily amenable to interactive exploration using suitable computer interfaces. For example, a click on an interesting location on the cluster display (Figure 13) might highlight the corresponding location on the indicator displays of Figure 15.

11.2 Case Study: Country Risk Ratings

Similar displays can be created to illustrate any data set. Another economic data set that may be of interest to economists is a set of nine indicators published by the Euromoney magazine (March 1996): Country risk ratings. In the display shown in Figure 16, like in the previous display of welfare and poverty, Finland is situated in the top left corner, together with, e.g., most West-European countries.
These kinds of displays provide an overview of the state of the world at a given moment, and a possibility to explore the state further. It would of course be of interest also to follow the changes in the state for several years. The Self-Organizing

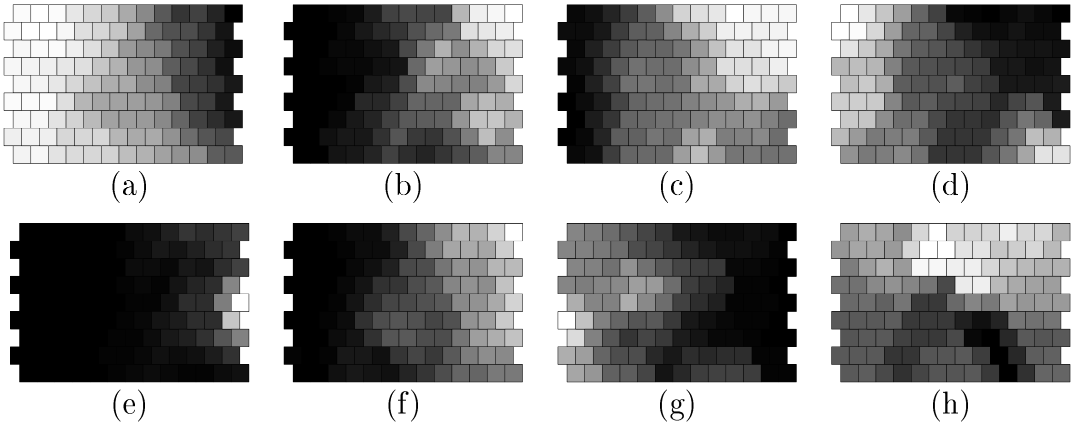


Figure 15: The values of some of the indicators visualized on the SOM groundwork. Since the countries have been organized into a natural order, the displays have clear "patterned" outlook instead of being purely random. Therefore they can be interpreted quickly. (a) Life expectancy at birth (years); (b) Adult illiteracy (%); (c) Share of food in household consumption (%); (d) Share of medical care in household consumption (%); (e) Population per physician; (f) Infant mortality rate (per thousand live births); (g) Tertiary education enrollment (% of age group); and (h) Share of the lowest-earning 20 percent in the total household income. In each display, white indicates the largest value and black the smallest, respectively.

Map is readily amenable also for such studies.

11.3 Conclusions

The SOM has been applied to case studies to show how it can be used as a decision-support system, to get a quick but yet quite accurate impression of the structures inherent in any data set.
Following exactly the same procedures the SOM could also be used for analyzing and visualizing sets of statistical indicators in other similar applications. For instance, the method has already been used for the analysis of states of banks [4]. The SOM formed a "solvency map," from which the state of the banks could be inferred at a glance. In time series analysis it is important that the nature of change in the state of the banks can be visualized on the map (e.g., as a slow shift toward the bankrupt region) even if the changes could not be predicted by more traditional methods.

References

[1] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery: An overview. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 1–34. AAAI Press / MIT Press, Menlo Park, CA, 1996.

[2] J. Iivarinen, T. Kohonen, J. Kangas, and S. Kaski. Visualizing the clusters on the self-organizing map. In C. Carlsson, T. Järvi, and T. Reponen, editors, *Proceedings of the Conference on Artificial Intelligence Research in Finland*,

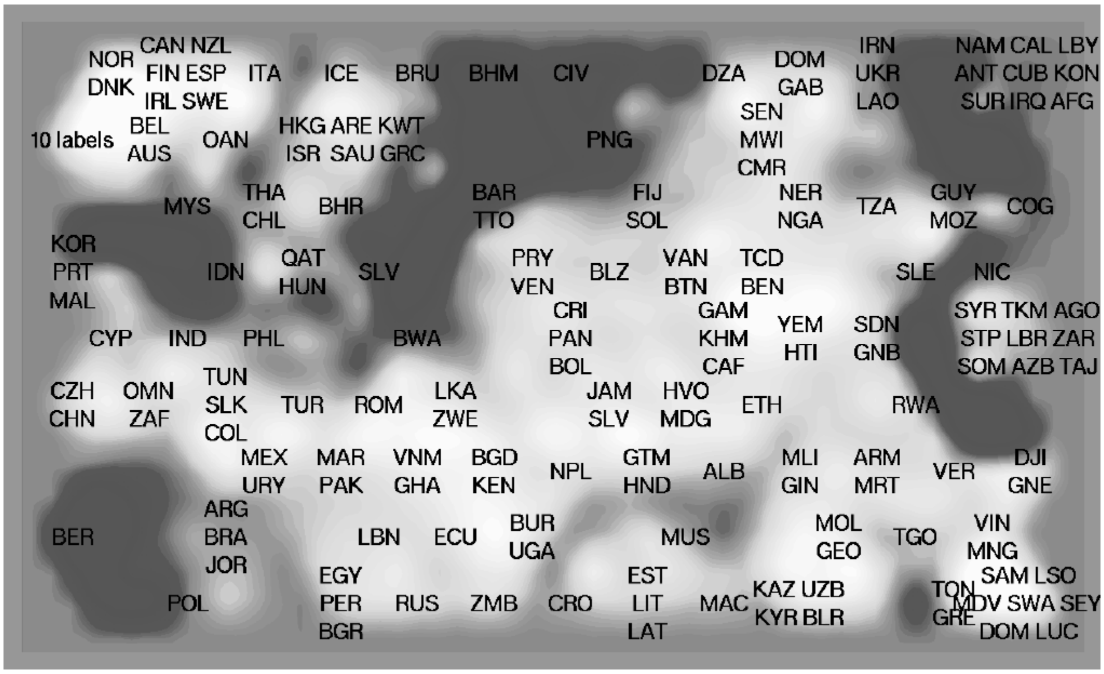


Figure 16: Country risk ratings (Euromoney, March 1996). An illustration of the structures within a data set that describes different aspects of the country risk (economic performance, political risk, debt, access to finance etc.). Based on the distribution of the original indicators on the map groundwork (not shown) it may be summarized that, e.g., the countries in the upper right hand corner have significant amounts of debts, and countries in the lower right hand corner perform poorly economically. Countries in the upper left hand corner perform best on every indicator. (The "10 labels" in the image refers to countries LUX, CHE, SGP, JPN, USA, NLD, DEU, AUT, GBR, and FRA.)

number 12 in Publications of the Finnish Artificial Intelligence Society, pages 122–126. Finnish Artificial Intelligence Society, Helsinki, Finland, 1994.

[3] A. K. Jain and R. C. Dubes. *Algorithms for clustering data.* Prentice Hall, Englewood Cliffs, NJ, 1988.

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

[5] J. W. Sammon, Jr. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18:401–409, 1969.

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

[7] World Bank. *World Development Report 1992.* Oxford Univ. Press, New York, NY, 1992.

12 Methods for Interpreting Self-Organized Maps in Data Analysis

Samuel Kaski, Janne Nikkilä, and Teuvo Kohonen

The Self-Organizing Map (SOM) can be used for forming overviews of multivariate data sets and for visualizing them on graphical map displays, as described in Section 11. Each map location represents certain kinds of data items and the value of a variable in the representations can be visualized in the corresponding locations on the map display. Examples of such component plane displays have been shown in Figure 15 in Section 11. The component planes contain all the information needed for interpreting the map but information about the relations of the variables remains implicit. We have developed methods that visualize explicitly the contribution of each variable in the organization of the map at different locations.
It is additionally possible to summarize the characteristics of different areas on the SOM, for instance areas corresponding to different clusters, by measuring the contribution of each variable in the cluster structure within the area.
We are currently in the process of evaluating the proposed methods in case studies. Here the methods will be demonstrated with a simple data set consisting of 13 properties of 16 animals. Each variable has the value one if the animal has the property and zero if it does not. A SOM of the animal data set is shown in Fig. 17. Different regions of the map represent different kinds of animals in an ordered fashion.

12.1 Local Factors

The SOM can be thought of as a nonlinear lattice of points that are determined by the model vectors in the high-dimensional data space. It is not possible to interpret the nonlinear lattice as simply as for example the set of linear factors obtained by factor analysis. The lattice can, however, be approximated *locally* by a linear hyperplane which is fitted to represent the model vectors within a certain radius on the map. The approximation can be computed with the principal component analysis algorithm resulting in two *local factors*.
The combined contribution of a variable on the local factors, computed as the sum of squares of the "factor loadings", at each location of the map lattice can be visualized as a gray-level display that resembles a component plane (Fig. 17**b**). It can be seen in the figure that the variable "has hair" contributes strongly to the organization of the map along a stripe in the middle of the map where the representation changes from birds to other animals. The variable "has hooves" contributes to the organization in the top right corner.

12.2 Summary Generation

The methods described above aim at making the basis of organization of the SOM explicit. They do not, however, further reduce the amount of data, and we have therefore developed a method for generating briefer summaries of the important characteristics of the maps. In this study the method is used in a partly manual mode but most of the steps can be automated.

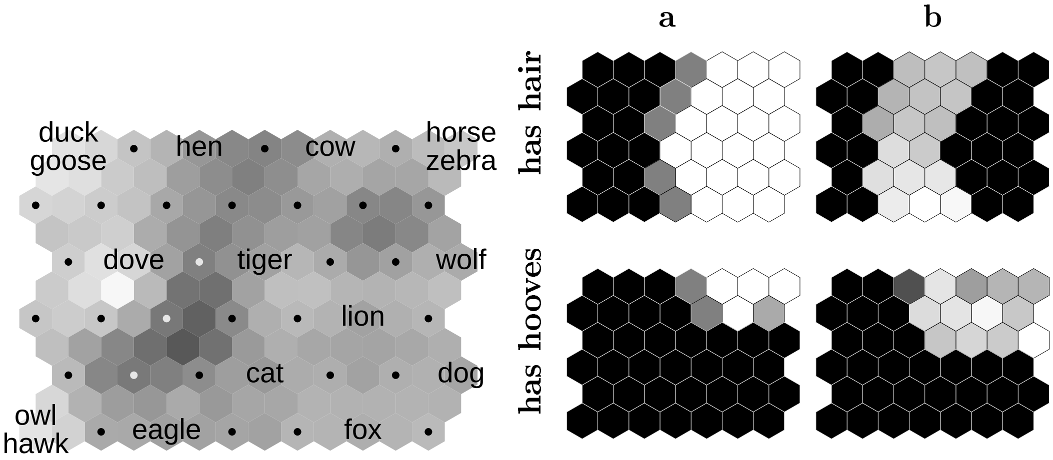


Figure 17: Sample illustrations of the methods applied to the animal data. The SOM of the animal data set is shown on the left; gray shades indicate clustering tendency (white: clustered area, dark: sparser area in between clusters). On the right, the top row visualizes the variable "has hair" and the bottom row "has hooves", respectively. **a** The component planes. Each plane describes the values of one variable at each location on the map. **b** The contribution of the variables in the two local factors (white: maximal contribution, black: minimal contribution).

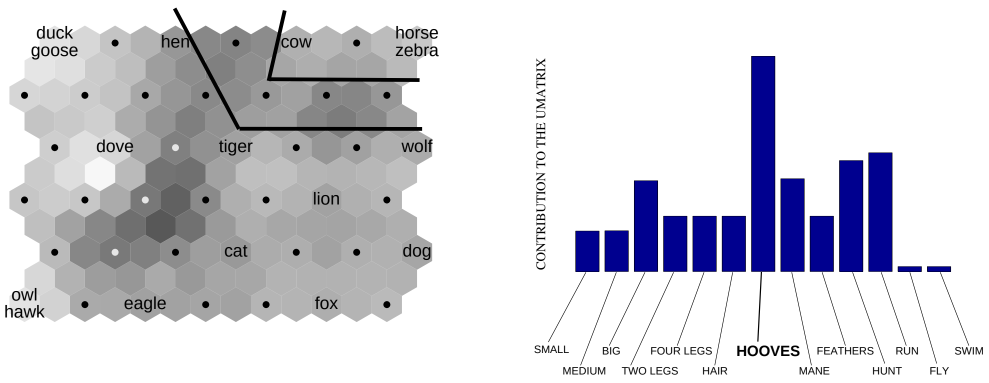


Figure 18: Characterization of a cluster in terms of the contributions of the original variables in the cluster structure. The region around the cluster in the top right corner is shown on the left, and the contribution of the variables within this area is shown on the right.

After the user has found some interesting area on the map, for example a cluster, we aim at summarizing *which of the original variables contributes most to the direction of the map around the area.* A very simple and easily computable measure of such contribution is the share of the component in the distances between neighboring map units around the cluster.

An example of the analysis of a clustered area is shown in Figure 18. The cluster consisting of cow, horse, and zebra seems to be characterized mainly by the variable "has hooves".

13 Coloring that Reveals High-Dimensional Structures in Data

Samuel Kaski, Jarkko Venna, and Teuvo Kohonen

When illustrating statistical tables, it is commonplace to visualize different groups of data items with different colors. For example, the World Bank visualizes different groups of economies, viz. low-income, middle-income, and high-income economies, by coloring the countries in each of the three groups with different, manually chosen colors on a world map display.

Similar visualizations are being used so pervasively that the question naturally arises, whether it would be possible to construct such a coloring that the relations of the colors would represent the relations of the clusters or, more generally, so that the *perceptual relations in the colors would reflect the relations between the high-dimensional data items.*

In Section 13.2 we present a solution in the special case that is especially useful for exploratory data analysis: coloring of data sets organized on Self-Organizing Maps. First, however, the basic setting is introduced in a simpler form in which the coloring is chosen interactively.

13.1 Simple method for interactive coloring

The starting point of the coloring is a Self-Organizing Map of the data set. The map can be used to visualize cluster structures in the data on a map display as discussed in Section 11. A sample display that describes the structures of welfare and poverty in the countries of the world is shown in Figure 13 of Section 11, and reproduced in Figure 19.

In the interactive coloring system the user decides, based on the clustering display, how many clusters there are in the data, points out the cluster centers, and chooses colors for them. A sample choice of the centers has been shown in Figure 19. An automatic system [1] may be used to make preliminary choices.

After the cluster centers have been colored, the color is "spread" to the neighborhood of each center; while the color spreads its intensity diminishes according to the distance it has passed. The clustering structure (illustrated in shades of gray in Figure 19) is taken into account when computing the distance: in the clustered areas the intensity diminishes more slowly than in the "ravines" between the clusters. Each location on the map receives the color that is a mixture of the colors that have spread to it from the cluster centers, each weighted by the distance of that unit from the corresponding center.

In the resulting map display (Figure 20) the colors have an intuitive interpretation: Each bright (pure) color corresponds to a certain data type, and mixed colors correspond to intermediate forms.

13.2 Automatic coloring

The coloring described in the previous section was more faithful to the relations between the actual data items than a completely manual coloring, but it was still

not perfect.
Ideally, the colors should be such that the relations between the high-dimensional data items would be reflected as closely as possible in the perceptual relations between their colors. The perceptual differences between colors can be approximated by distances in a color space called CIELab. Therefore, any coloring actually corresponds to a certain mapping of the high-dimensional data set into a smaller-dimensional space, the CIELab color space.
The perceptual color space is unfortunately only three-dimensional, and therefore it is impossible in general to construct a mapping that would preserve *all* of the pairwise distances between the data items. Fortunately, all of the distances are not equally important for the quality of the coloring. Longer distances do not usually need to be represented as accurately as shorter ones. In traditional hierarchical clustering, for example, the relation between data items is represented as their relation in a hierarchical tree of clusters. Distances of items in different clusters will then not be represented individually but in terms of the relation between the clusters.
It has turned out in our studies that it is possible to construct a coloring that concentrates almost entirely on representing the local, small distances, but which still becomes globally ordered. The only additional constraint needed is that no data item that is originally farther away may attain a color that is, intuitively speaking, in between the colors of two close-by data items. Such a color would break the global order. These constraints have turned out flexible enough to allow the mapped colors

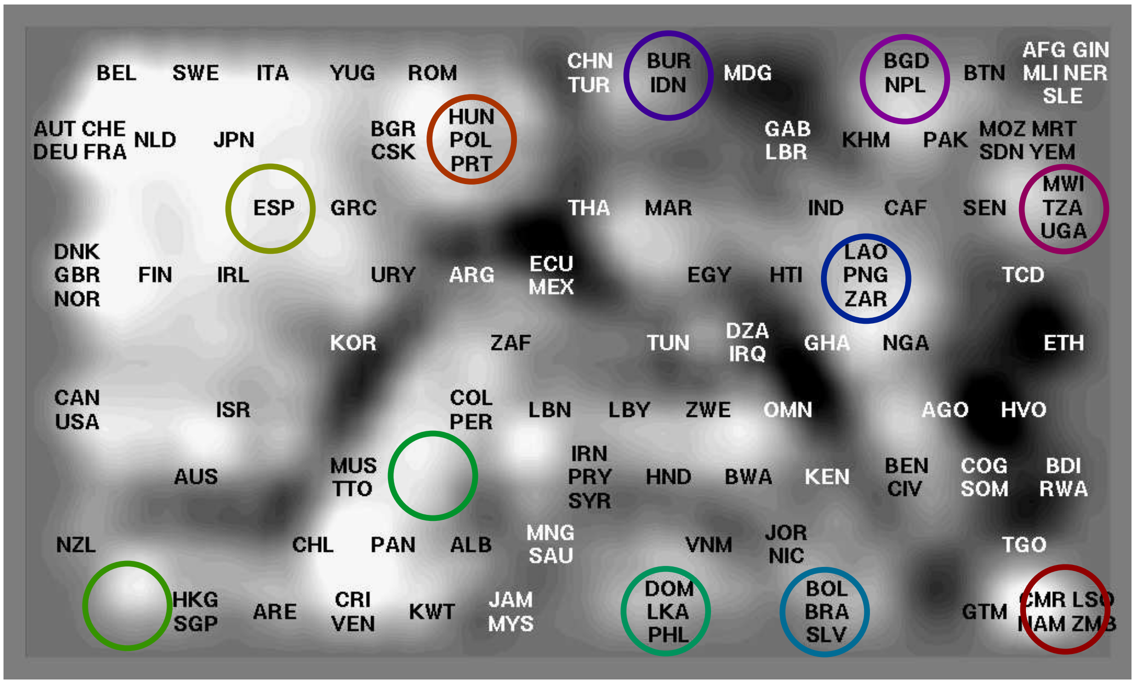


Figure 19: Locations of the chosen cluster centers, shown on top of the clustering diagram formed by the Self-Organizing Map algorithm. The diagram describes different aspects of the welfare and poverty of different countries. Light areas represent areas of a high degree of clustering and dark areas gaps in the degree of clustering. Different types of welfare and poverty are visible as the clustered (light) areas on the map, separated by the dark "ravines".

to fill the available color space reasonably well.
The mapping method has been applied to coloring SOM displays which are well-suited for such coloring. Neighboring map units represent similar data items, and therefore the distances between neighboring units may be regarded as the local ones that will be represented accurately.
The mapping is constructed by requiring that (1) the local distances, i.e. distances between model vectors of neighboring map units, will be preserved as accurately as possible; (2) the model vectors belonging to farther-away map units remain farther away (but their relative order may be arbitrary); and (3) the colors remain within the region of the color space which is representable in the chosen media, for example by the CRT tube. Each of these three conditions was dressed into a term in a cost function, whereafter the minimum of the cost function can be sought with any standard optimization algorithm. Sor far we have used stochastic gradient descent which aids in avoiding local minima.
The result of mapping the model vectors of the SOM of Figure 19 into the CIELab color space is shown in Figure 21. When the map units were colored according to the projection (Figure 22), the differences in the hue of neighboring map units corresponded well with the distances in the original data space, depicted as shades of gray in Figure 19. In addition, the hues became ordered globally; different clustered areas attained different, relatively uniform hues.
The resulting coloring is almost tailored for the human color vision system which

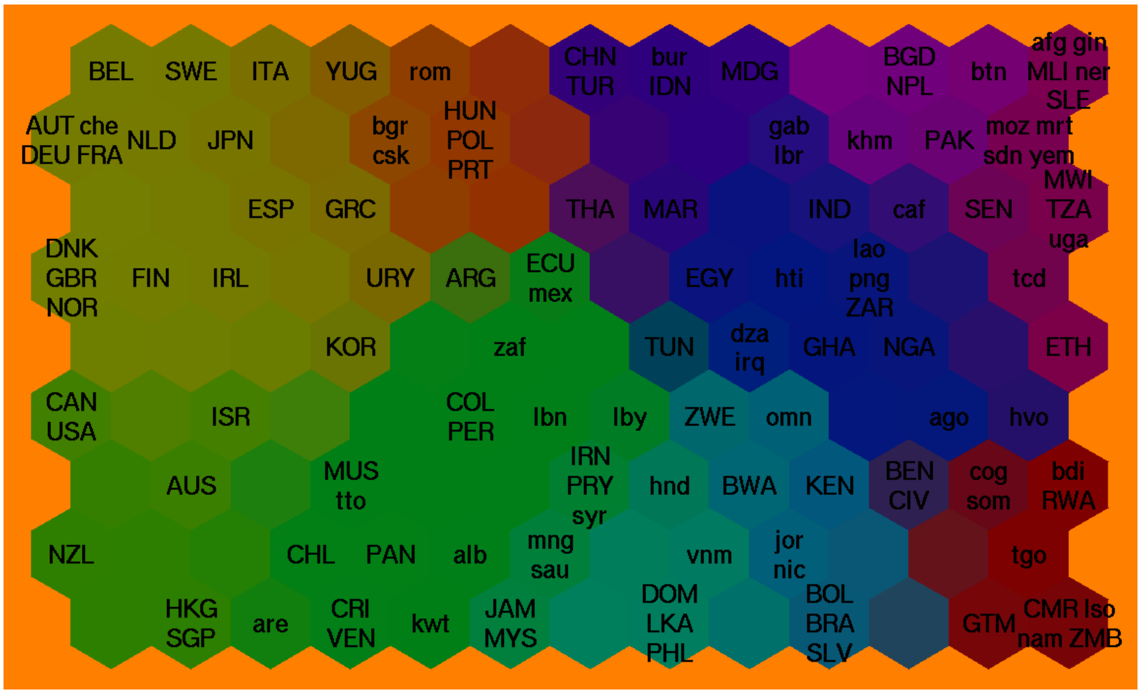


Figure 20: A Self-Organizing Map display in which different, manually chosen clusters have been colored with different colors. The colors are brightest in the cluster centers, and change gradually with increasing distance from the center. The colors change in proportion to the clustering structure so that tight clusters have a relatively homogeneous coloring, and the color changes the more sharply the more clear-cut the border between the clusters is.

is very accurate in detecting differences between the colors of neighboring areas, in this case the neighboring map units.

Relation to possible alternative methods. The traditional multidimensional scaling methods like the Sammon's mapping [2] could in principle be used for projecting the data items into the color space. They do not, however, produce as flexible mappings as our method, since they try to represent *all* of the pairwise distances. The local differences will then necessarily be represented less accurately, except in special cases. Moreover, it may be more difficult to utilize all of the available color space when the mapping is more stiff.
In principle our method could be used to map the data set directly, instead of mapping the model vectors of a SOM computed from the data set. It would, however, be more difficult to define which distances are local enough so that they should be represented accurately. If it is necessary to obtain a characteristic color for each data item then local linear approximations, for instance, may be used to complement the mapping.

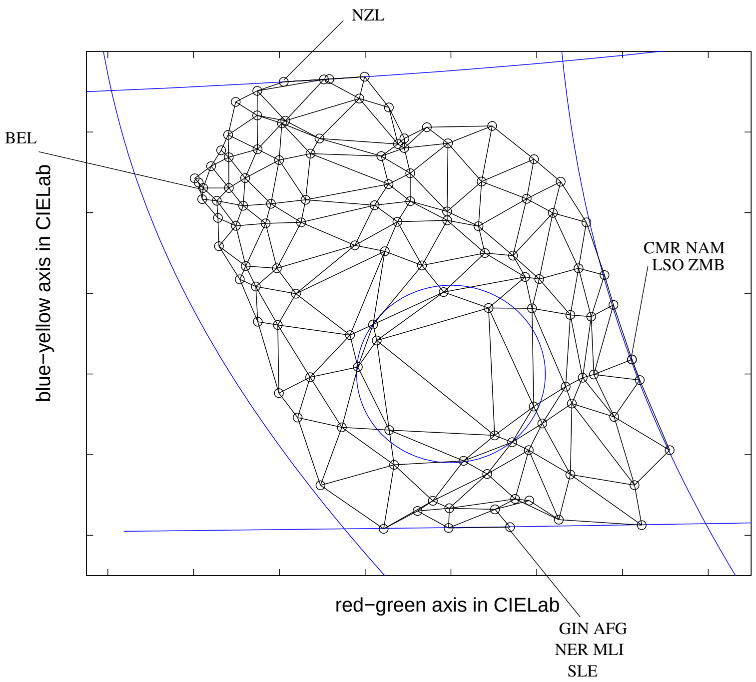


Figure 21: The projection of the model vectors of the SOM of Figure 19 into the CIELab color space. Only part of the space was used to ensure that the obtained colors would differ only in one perceptual quality, the hue. The lightness was fixed to a constant value, which reduces the space into a two-dimensional slice of the original three-dimensional color space, and projections onto non-saturated colors in the middle, encircled in the figure, were discouraged. The small circles denote the projections of the model vectors. Projections of model vectors of neighboring map units have been connected with lines. The long lines delimit the region representable by a typical CRT tube.

13.3 Example: coloring of the world map according to poverty types

The coloring can be even more useful if the original data set can be visualized also in some other manner. Then each data item can be colored with the color that the item has on the SOM display. The welfare and poverty structures can be visualized in a straightforward manner: the countries can be colored according to their welfare or poverty type on a geographic map display (Figure 23).

The result is a display where countries having a similar welfare or poverty type have been colored similarly irrespective of their geographical location. Japan and Australia, for example, are fairly similar to the European countries and the USA and Canada. Countries which belong to very different types than their neighbors pop out strongly, like Japan, Sri Lanka, and Albania.

Visualizations like the one shown in Figure 23 can be very useful if the data has a "natural" ordering like the geographical order here. In fact, *any* order of the data items can be used. For example, if the countries were ordered simply according to the GNP per capita in a statistical table and colored using a SOM, then countries in which the welfare or poverty type is different from the other countries having a similar value of GNP per capita would be clearly discernible based on sharp discontinuities in the coloring.

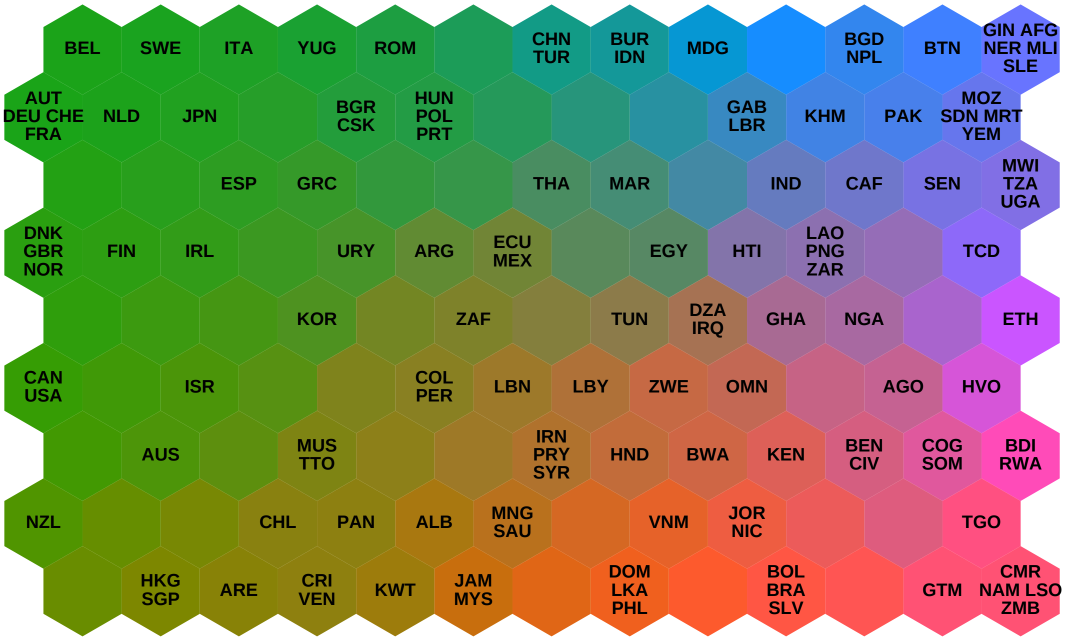


Figure 22: Coloring of the SOM according to the projection shown in Figure 21. The relative differences in the colors of the neighboring map units reflect closely the clustering display in Figure 19. The cluster areas have relatively uniform coloring, and the differences are the larger the steeper the "ravine" between the clusters is.

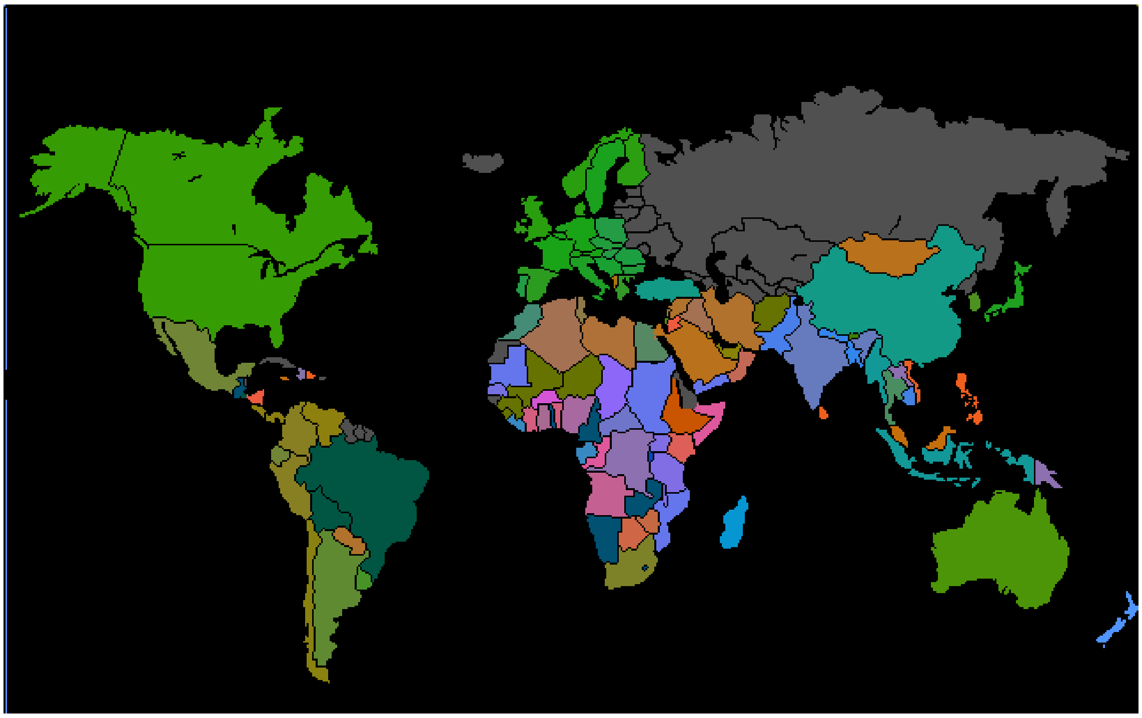


Figure 23: The types of welfare and poverty that the Self-Organizing Map has revealed can be visualized on a geographical world map. Each country is colored according to its color on the SOM display (Figure 22). Countries for which no data was available (like Russia) have been colored with dark gray.

13.4 Conclusions

We have constructed an automatic method for coloring data so that the perceptual properties of the coloring reflect closely the properties of the high-dimensional statistical data. The easily interpretable coloring makes it possible to visualize complex statistical structures automatically for non-experts in statistics.

References

[1] S. Kaski, J. Venna, and T. Kohonen. Tips for processing and color-coding of self-organizing maps. In Guido Deboeck and Teuvo Kohonen, editors, *Visual Explorations in Finance with Self-Organizing Maps*, pages 195–202. Springer, London, 1998.

[2] John W. Sammon, Jr. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18:401–409, 1969.

14 Self-Organization of Very Large Document Collections

Teuvo Kohonen, Samuel Kaski, Krista Lagus, Timo Honkela, Jarkko Salojärvi, Jukka Honkela, Vesa Paatero, Antti Saarela, and Antti Ahonen

In the vast majority of SOM applications, the input data constitute high-dimensional real *feature vectors*. In the SOMs that form similarity graphs of *text documents*, models that describe collections of words in the documents may be used. The models can simply be weighted histograms of the words regarded as real vectors, but usually some dimensionality reduction of the histograms is carried out, as we shall see next.

14.1 Statistical models of documents

14.1.1 The primitive vector space model

In the basic *vector space model* [1] the stored documents are represented as real vectors in which each component corresponds to the frequency of occurrence of a particular word in the document: the model or document vector can be viewed as a weighted word histogram. For the weighting of a word according to its importance one can use the Shannon entropy over document classes, or the inverse of the number of the documents in which the word occurs ("inverse document frequency"). The main problem of the vector space model is the large vocabulary in any sizable collection of free-text documents, which means a vast dimensionality of the model vectors.

14.1.2 Latent semantic indexing (LSI)

In an attempt to reduce the dimensionality of the document vectors, one often first forms a matrix in which each column corresponds to the word histogram of a document, and there is one column for each document. After that the factors of the space spanned by the column vectors are computed by a method called the singular-value decomposition (SVD), and the factors that have the least influence on the matrix are omitted. The document vector formed of the histogram of the remaining factors has then a much smaller dimensionality. This method is called the *latent semantic indexing (LSI)* [2].

14.1.3 Randomly projected histograms

It has been shown experimentally that the dimensionality of the document vectors can be reduced radically by a random projection method [3], [4] without essentially losing the power of discrimination between the documents. Consider the original document vector (weighted histogram) $\mathbf{n}_i \in \Re^n$ and a rectangular random matrix $\mathbf{R}$, the elements in each column of which are assumed to be normally distributed. Let us form the document vectors as the *projections* $\mathbf{x}_i \in \Re^m$, where $m \ll n$:

$$\mathbf{x}_i = \mathbf{R}\mathbf{n}_i \ . \tag{78}$$

It has transpired in our experiments that if m is at least of the order of 100, the similarity relations between arbitrary pairs of projection vectors $(\mathbf{x}_i, \mathbf{x}_j)$ are very good approximations of the corresponding relations between the original document vectors $(\mathbf{n}_i, \mathbf{n}_j)$, and the computing load of the projections is reasonable; on the other hand, with the radically decreased dimensionality of the document vectors, the time needed to classify a document is radically decreased.

14.1.4 Histograms on the word category map

In the "self-organizing semantic map" method [5] the words of free natural text are clustered onto neighboring grid points of a special SOM. Synonyms and closely related words such as those with opposite meanings and those forming a closed set of attribute values are often mapped onto the same grid point. In this sense this clustering scheme is even more effective than the thesaurus method in which sets of synonyms are found manually.

The input to the "self-organizing semantic map" usually consists of adjacent words in the text taken over a moving window. Let a word in the vocabulary be indexed by k and represented by a unique random vector $\mathbf{r}_k$. Let us then scan all occurrences of word (k) in the text in the positions $j(k)$, and construct for word (k) its "average context vector"

$$\mathbf{x}_k = \begin{bmatrix} \mathrm{E}\left\{\mathbf{r}_{j(k)-1}\right\} \\ \varepsilon\, \mathbf{r}_{j(k)} \\ \mathrm{E}\left\{\mathbf{r}_{j(k)+1}\right\} \end{bmatrix} , \tag{79}$$

where E means the average over all $j(k)$, $\mathbf{r}_{j(k)}$ is the random vector representing word (k) in position $j = j(k)$ of the text, and ε is a scaling (balancing) parameter. Notice that this expression has to be computed only once for each different word, because the $\mathbf{r}_{j(k)}$ for all the $j = j(k)$ are identical.

In making the "semantic SOM" or the *word category map*, all the $\mathbf{x}_k$ from *all the documents* are input iteratively a sufficient number of times. After that each grid point is labeled by *all those words (k)*, the $\mathbf{x}_k$ of which are mapped to that point. In this way the grid points usually get multiple labels. A sample map is shown in Figure 24.

In forming the "word category histogram" for a document, the words of the document are scanned and counted at those grid points of the SOM that were labeled by that word. In counting, the words can be weighted by the Shannon entropy or the inverse of the number of documents in the text corpus in which this word had occurred (= "inverse document frequency").

The "word category histograms" can be computed reasonably fast, much faster than, e.g., the LSI.

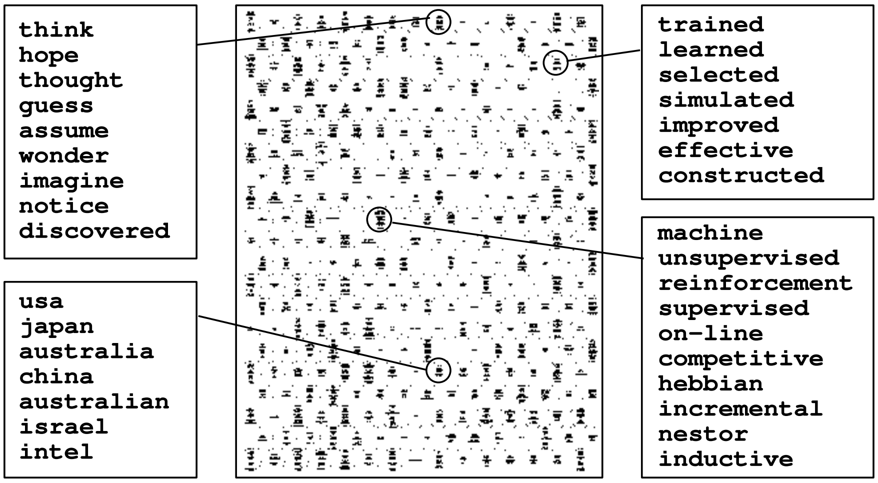


Figure 24: Examples of some clear "categories" of words on the word category map of the size of 15 by 21 nodes. The word labels of the map nodes have been shown with a tiny font on the map grid, and four nodes have been enlarged in the insets.

14.1.5 Randomly projected word category histograms

In a great number of experiments performed by us it has transpired that if the histograms on the word category maps are used as models, the ability of our method to discriminate between the documents is reduced if the grid points in the word category map contain more than, say, ten words on the average: specific information contained in the words is then lost. We have been interested in very large document collections that may contain, say, hundreds of thousands of unique words, and even after discarding very rare words, the remaining vocabulary may consist of tens of thousands of words. In order to keep the number of words on each point of the word category map at the tolerable level, the word category map therefore had to be reasonably large, for example 13,432 grid points in some of our latest experiments. The histograms of this dimensionality we then again projected randomly to form 315-dimensional statistical document vectors.
The combination of word categorization and random projection guarantees a certain degree of invariance with respect to the choice of, e.g., synonyms, while a high degree of discrimination between documents can still be maintained, for similar reasons as in the random projection method.

14.1.6 Construction of the random projections by pointers

There exists a special method for forming projections that gives as good results as the random projections discussed above but is computationally much more efficient for sparse input vectors (Table 5). The method has been discussed in Section 15.

14.2 Construction of the document map

Our original document-organization system named the WEBSOM (`http://websom.hut.fi/websom/`) used word-category histograms as statistical models of the documents. Certain reasons, among them the accuracy of classification, have recently led us to prefer the straightforward random projection (or its shortcut computation by the pointers) in forming the statistical models of the documents. We have carried out numerous experiments with maps of very different sizes; results from a sample comparison have been given in Table 5. In these experiments the word category map had 1598 grid points, and the dimension of the projected model was 270.

Table 5: Classification accuracies in a sample comparison test

	Matrix product	Pointer method (3 pointers/column)
Random projection	68.0	67.5
Randomly projected word category histogram	66.0	67.0

It must also be taken into account that with the word category map method we have to deal with an extra self-organizing process, whereas forming the random projection is a straightforward computation.
Our current method is a collection of programs that can be combined in different ways. A brief overview of the computing phases is given in the following.

Preprocessing. From the raw text, nontextual and otherwise nonrelevant information (punctuation marks, articles and other stopwords, message headers, URLs, email addresses, signatures, images, and numbers) was removed. The most common words, and words occuring rarely (e.g., less than 50 times in the corpus) were also discarded. Each remaining word was represented by a unique random vector of dimensionality 90.
For a language like Finnish that has plenty of inflections, we have used a *stemmer*. In our experiments we have so far regarded the various English word forms as different "words" in vocabulary, but a stemmer could be used for English, too.

Formation of statistical models. To reduce the dimensionality of the models, we have used both randomly projected word category histograms and randomly projected word histograms, weighted by the Shannon entropy or "inverse document frequency."

Formation of the document map. The document maps were formed automatically by the SOM algorithm, for which the statistical models of documents were used as input. The size of the SOM was determined so that on the average 10 to 15 articles were mapped onto each grid point; this figure was mainly determined for the convenience of browsing.
The speed of computation, especially of large SOMs can be increased by several methods. In our latest experiments we have used the Batchmap algorithm (discussed in Section 1) in which the winner search was accelerated. The search was started in

the neighborhood of corresponding winners at the last cycle of iteration (discussed in Section 9). The distance computation, or in this case actually computation of inner products, can be speeded up even further by neglecting the components having the value zero; a large proportion of the components are zeroes since the input vectors are still sparse after the dimensionality has been reduced by the random projection with pointers.

The size (number of grid nodes) of the maps was increased stepwise during learning using the estimation procedure discussed in Section 9. After each increase the winners for all data vectors can be found quickly by utilizing the estimation formula that was used in increasing the map size, equation 75 in Section 9. The winner is the map unit for which the inner product with the data vector is the largest, and the inner products can be computed rapidly using the expression

$$\mathbf{x}^T\mathbf{m}_h^{(d)} = \alpha_h\mathbf{x}^T\mathbf{m}_i^{(s)} + \beta_h\mathbf{x}^T\mathbf{m}_j^{(s)} + (1-\alpha_h-\beta_h)\mathbf{x}^T\mathbf{m}_k^{(s)} \; . \tag{80}$$

Here d refers to model vectors of the large map and s of the small map, respectively. The expression (80) can be interpreted as the inner product between two *three-dimensional* vectors, $[\alpha_h; \beta_h; (1-\alpha_h-\beta_h)]^T$ and $[\mathbf{x}^T\mathbf{m}_i^{(s)}; \mathbf{x}^T\mathbf{m}_j^{(s)}; \mathbf{x}^T\mathbf{m}_k^{(s)}]^T$, *irrespectively of the dimensionality of* $\mathbf{x}$. If necessary, the winner search can be speeded up even further by restricting the search to the area of the larger map that corresponds to the neighborhood of the winner on the smaller map.

For very large maps it may be necessary to save the amount of memory needed for storing the maps. We have represented the model vectors with reduced accuracy and used probabilistic updating to maintain numerical accuracy in adapting the model vectors.

User interface. The document map was presented as a series of HTML pages that enable exploration of the grid points: when clicking the latter with a mouse, links to the document data base enable reading the contents of the articles. Depending on the size of the grid, subsets of it can first be viewed by zooming. Usually we use two zooming levels for bigger maps before reading the documents.

There is also an automatic method, discussed in Section 16, for assigning descriptive signposts to map regions; in deeper zooming, more signs appear. The signposts are words that appear often in the articles in that map region and rarely elsewhere.

Content-addressable search. The HTML page can be provided with a form field into which the user can type an own query in the form of a short “document.” This query is preprocessed and a document vector (histogram) is formed in the same way as for the stored documents. This histogram is then compared with the “models” of all grid points, and a specified number of best-matching points are marked with a round symbol, the diameter of which is the larger, the better the match is. These symbols provide good starting points for browsing.

If the document map is very large, the comparison between the document vector and all the model vectors is time-consuming. It is, however, possible to make rapid approximations by restricting the comparisons in such subspaces of the original space that best represent the (local) organization of the map.

A problem may be encountered if the user wants to use a single keyword or a few keywords only as a “key document.” Such queries make very bad “histograms.” In this case it is more advisable to use *two different modes of use* of the WEBSOM: the user must then specify whether a document-type or keyword-type query has to be used. In the former case the operation is like described before; in the latter case one has to index each word of the vocabulary by pointers to those documents where these words occur, and use a rather conventional indexed search to find the matches.

14.3 Examples

14.3.1 The largest published map

The biggest document map we have published so far consists of 104,040 grid points. Each model is 315-dimensional, and has been made by projecting a word category map with 13,432 grid points randomly onto the 315-dimensional space. The text material was taken from 80 very different Usenet newsgroups and consisted of 1,124,134 documents with average length of 218 words. The size of the finally accepted vocabulary was 63,773 words. The words were weighted by the Shannon entropy computed from the distribution of the words into 80 classes (newsgroups). It took about 1 month to process the two SOMs without our newest speedup methods; searching occurs in nearly real time.
The accuracy of classifying a document into one of the 80 groups was about 80 per cent.
Fig. 1 exemplifies a case of content-addressable search. The document map has been depicted in the background, and the shades of gray correspond to document clusters. The 20 grid points, the models of which matched best with the short query, are visible as a small black heap on the left-hand side of the document map. Using a browser, the documents mapped to grid points of the document map can be read out from the HTML page. Two title pages are shown.
Actually there is only one article in Fig. 1 that deals with NN chess. However, the other computer chess documents were so similar that they were returned, too. About one fourth of the found documents obviously does not deal with chess.

14.3.2 The largest map being processed

We are currently finishing the computation of a map of all of the patent abstracts in the world that are available in electronic form, about 7,000,000 in total. The map consists of about 1,000,000 units.

14.4 Conclusions

We have demonstrated that it is possible to scale up the SOMs in order to tackle very large-scale problems. Additionally, it has transpired in our experiments that the encoding of documents for their statistical identification can be performed much more effectively than believed a few years ago [2]. In particular, the various random-projection methods are as accurate in practice as the ideal theoretical vector space method, but much faster to compute than the eigenvalue methods (e.g., LSI) that have been used extensively to solve the problem of large dimensionality.
The content-addressable search must obviously be implemented differently when complete new “documents” are used as key information vs. when only a few key-

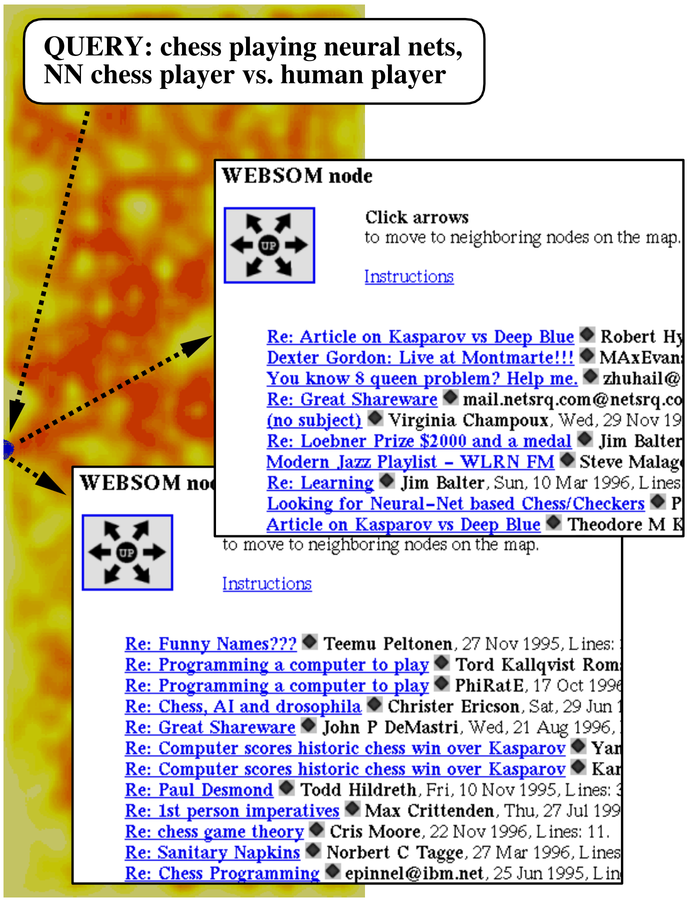


Figure 25: Content-addressable search from a 1,124,134-document WEBSOM

words are used. To this end one must first identify the users' needs, e.g., whether background information to a given article is wanted, or whether the method is used as a kind of keyword-directed search engine.

Finally it ought to be emphasized that the order that ensues in the WEBSOM may not represent any taxonomy of the articles and does not serve as a basis for any automatic indexing of the documents; the similarity relationships better serve "finding" than "searching for" relevant information.

References

[1] Salton G, McGill MJ. *Introduction to modern information retrieval.* McGraw-Hill, New York, 1983

[2] Deerwester S, Dumais S, Furnas G, Landauer K. Indexing by latent semantic analysis. *J Am Soc Inform Sci*, 1990; 41:391-407

[3] Kaski S. Data exploration using self-organizing maps. Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series No 82, 1997. Dr Tech Thesis, Helsinki University of Technology, Finland

[4] Kaski S. Dimensionality reduction by random mapping. In: Proc of IJCNN'98, Int Joint Conf on Neural Networks. IEEE Press, Piscataway, NJ, 1998, pp 413-418

[5] Ritter H, Kohonen T. Self-organizing semantic maps. *Biol Cyb*, 1989; 61:241-254

15 Construction of Random Projections of Word Histograms by Pointers

Teuvo Kohonen

In the basic *vector space model* [1], documents were represented statistically as real vectors in which each component corresponds to the frequency of occurrence of a particular word in the document: the model or document vector can be viewed as a weighted word histogram. The main problem of the vector space model is the large vocabulary in any sizable collection of free-text documents, which means a vast dimensionality of the model vectors.

It was shown previously that the dimensionality of the document vectors can be reduced radically by a random projection method [2,3] without essentially losing the power of discrimination between the documents.

Before description of the new encoding of the documents, some experimental results are presented that motivate its idea. Table 6 compares a few projection methods in which the model vectors, except in the first case, were always 315-dimensional. As the material in this experiment we used 18,540 English documents from 20 Usenet newsgroups of Internet. When the text was preprocessed as explained in Sec. 14, the remaining vocabulary consisted of 5,789 words or word forms. When the *document map* as discussed more closely in Sec. (*ibid*) was formed, each document was mapped onto one of its grid points. These points were then classified according to the majority of newsgroup names in them. All documents that represented a minority group at any grid point were counted as classification errors.

The classification accuracy of 68.0 per cent reported on the first row of Table 6 refers to a classification that was carried out with the vector-space model with full 5789-dimensional histograms as document vectors. In practice, this kind of classification would be orders of magnitude too slow.

Random projection of the original document vectors onto a 315-dimensional space yielded, within the statistical accuracy of computation, the same figures as the basic vector space method. This is reported on the second row. The figures are averages from seven statistically independent tests, like in the rest of the cases.

Consider now that we want to simplify the projection matrix in order to speedup computations. We do this by thresholding the matrix elements, or using sparse matrices. Such experiments are reported next. The following rows have the following meaning: Third row, the originally random matrix elements were thresholded to $+1$ or -1; fourth row, exactly 5 randomly distributed ones were generated in each column, and the other elements were zeroes; fifth row, the number of ones was 3; and sixth row, the number of ones was 2, respectively.

The sparse projection matrices have now turned out sufficiently good in producing reasonable classification accuracies, and we shall next concentrate on fast computation of the matrix-vector products. Consider first the following trivial-looking piece of pseudocode, where we form the product $\mathbf{x} = \mathbf{R}\mathbf{n}$:

Table 6: Classification accuracies of documents, in per cent, with different projection matrices $\mathbf{R}$. The figures are averages from seven test runs with different random elements of $\mathbf{R}$.

	Accuracy	Standard deviation due to different randomization of $\mathbf{R}$
Vector space model	68.0	–
Normally distributed $\mathbf{R}$	68.0	0.2
Thresholding to $+1$ or -1	67.9	0.2
5 ones in each column	67.8	0.3
3 ones in each column	67.4	0.2
2 ones in each column	67.3	0.2

```
for i:=1 step 1 until m do x(i):=0 ;
for all (i,j) such that R(i,j)=1 begin
      x(i):=x(i)+n(j) ;
end
```

This scheme is supposed to give us the idea that if we reserve a memory array for $\mathbf{x}$ that acts like an accumulator, another array for $\mathbf{n}$, and *permanent address pointers* from all the locations of the $\mathbf{n}$ array to all such locations of the $\mathbf{x}$ array for which the matrix element $R(i, j)$ of $\mathbf{R}$ is equal to one, we can form the product very fast by following the pointers.

In the method that is actually used we do not project ready histograms, but the pointers are already used with each word in the text in the construction of the low-dimensional document vectors. When scanning the text, the hash address for each word is formed, and if the word resides in the hash table, those elements of the $\mathbf{x}$ array that are found by the (say, three) address pointers stored at the due hash table location are incremented by the weight value of that word. The weighted, randomly projected word histogram obtained in the above way may be normalized optionally. The computing time needed to form the histograms in the above way is about 20 per cent of that of the usual matrix-product method. We have now some indication for the same speedup holding for larger maps, too.

References

[1] Salton G, McGill MJ. *Introduction to modern information retrieval.* McGraw-Hill, New York, 1983

[2] Kaski S. Data exploration using self-organizing maps. Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series No 82, 1997. Dr Tech Thesis, Helsinki University of Technology, Finland

[3] Kaski S. Dimensionality reduction by random mapping. In: Proc of IJCNN'98, Int Joint Conf on Neural Networks. IEEE Press, Piscataway, NJ, 1998, pp 413-418

16 Method for Characterizing Document Map Areas with Keywords

Krista Lagus and Samuel Kaski

When large collections of data are organized onto a map to visualize the collection, there is the need to characterize the different map areas. Characterization methods exist that can be used with any self-organizing maps (see Section 12), but with text document maps (see Section 14) there is a further possibility: keywords can be extracted from the documents and written on the map display to characterize the underlying area. The keywords aid in forming an overview of the document collection and ease interpretation of individual map areas. Furthermore, the keywords serve as navigation aids or *landmarks* during exploration of the map: they provide cues for maintaining a sense of location while moving along and across different zoom levels of the map display.

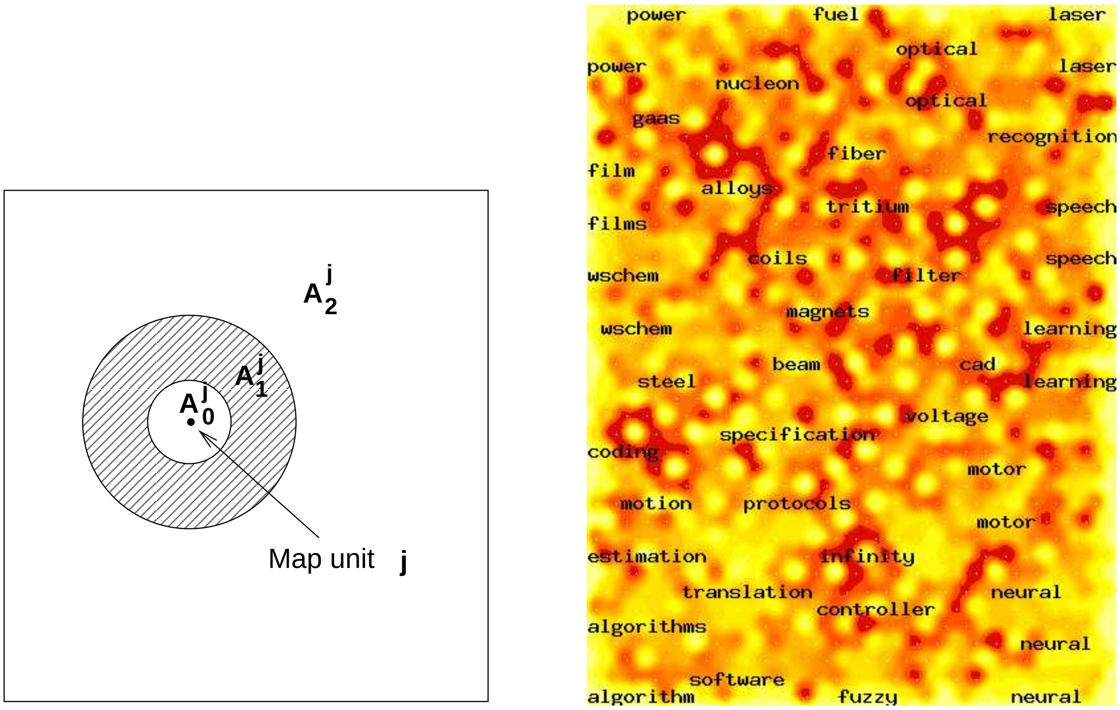
16.1 Keywords for map areas

A good descriptor of a cluster characterizes some outstanding property of the cluster in relation to the rest of the data collection. Therefore, when characterizing a cluster with a keyword, (1) the word must be outstanding within the cluster compared to other words in the cluster, and (2) the word must be relatively more outstanding in the cluster than elsewhere in the collection. These requirements can be combined into the following general form of a goodness measure G for word w in cluster k: $G(w,k) = F^{clust}(w,k) \times F^{coll}(w,k)$. By defining F^{clust} to describe the relative occurrence of word w in the cluster, and defining F^{coll} to relate the word to its occurrence in the rest of the collection, the obtained goodness measure implements our intuition of a good descriptor.

The same principle can be used to describe any map areas instead of clusters. However, the contents of map areas often change gradually without clear borders on the map. Therefore, instead of defining strict borders artificially, a better idea is to leave around each area to be characterized a "neutral zone" that neither supports nor inhibits a keyword (see Figure 26a). If the neutral zone is wide enough, it is probable that the topic clearly changes when moving over the zone. The goodness measure is described in detail in [1].

16.2 Labels for visual map displays

The keywords cannot be placed arbitrarily, however, since they may not overlap on the visual display, and since it is desirable that most areas obtain some characteristic label. Furthermore, the number of keywords should not be too large, to avoid overloading the visual display. To find for a map the optimal combination of N keywords, one would have to consider all the possible combinations of N keywords, which is in general prohibitively slow for larger maps. However, the following heuristic strategy may be used to obtain a good labeling in linear time: (1) Consider the proposed keywords in the *best-first order* determined by the goodness value G, and (2) accept a keyword as a label for the map display if the location associated with



a) b)

Figure 26: **a)** Only the white areas participate in calculating the goodness G of words in map unit j. The frequent occurrence of a word within the inner circle, area A_0^j, increases its possibility of becoming a keyword. Its frequent occurrence outside the outer circle, area A_2^j, inhibits the keyword. The shaded area A_1^j is disregarded and thus acts as a neutral zone neither giving support to the word nor inhibiting it. **b)** The top level map view of a document map that organizes a collection of 10,000 scientific abstracts from INSPEC database, labeled with the described method.

the keyword is far enough from already accepted labels. Labelings with varying density can be obtained for different map display levels (zoom levels) by controlling the size of the map area in calculation of G as well as the minimum distance between accepted labels.

Using this scheme we have obtained satisfying labelings for WEBSOM document maps, e.g., the one in Figure 26b. Further examples of such maps can be explored starting from the WWW page `http://websom.hut.fi/websom/`.

The applicability of the method is not limited to document maps. We believe the method could be used to obtain characterizations for maps of very different kinds of data, given that there exists some text material that can be associated with the data items and therefore with the map units.

References

[1] K. Lagus and S. Kaski. *Keyword selection method for characterizing text document maps.* Submitted to ICANN'99, Ninth Int. Conf. on Artificial Neural Networks.

17 Using Self-Organizing Maps for Natural Language Processing

Timo Honkela, Ville Pulkki, and Teuvo Kohonen

Development of large-scale natural language processing applications is restricted by quantitative and qualitative limitations. Quantitatively, a system for even a moderately narrow domain requires a substantial knowledge base. One suggested solution for this problem has been an approach where vast common repositories of knowledge items (frames, facts, rules) have been collected. Qualitatively problematic areas remain, e.g., graded phenomena, inherent ambiguity of natural language, and subjectivity and variation in natural language generation and interpretation. Gradual changes in the domain of the application make non-adaptive systems vulnerable.

The predominant approach among computerized models of language is based on predetermining and coding the linguistic categories and rules "by hand". The methodological basis is symbol manipulation. However, the fact that the expressions in natural language appear to be inherently symbolic and discrete does not imply that symbolic descriptions of linguistic phenomena are sufficient. This is expecially remarkable when semantic and pragmatic issues are considered, i.e., the ability of a system to interpret natural language expressions. To be able to model the gradually changing relation between continuous phenomena and discrete symbols, the building blocks of the theory must be sufficiently powerful.

The Self-Organizing Map [1] (SOM) algorithm can be used to automatically create implicit emergent categories from uncategorised linguistic input [2]. Our experiments have shown have unrestricted textual input can be analyzed by the SOM [3]. As the result a word category map is created. In the following, some of the details of the basic experiment are described.

The encoding of the input words was made using a 90-dimensional random real vector for each word. The codes were statistically independent so that there was no correlation between them. The code vectors of the words in the triplet, i.e., three subsequent words in the text, were then concatenated into a single input vector $\mathbf{x}(t)$, the dimensionality of which was thus 270. The 270-dimensional input vectors $\mathbf{x}(t)$ were used as inputs to the SOM algorithm. The SOM array itself was a planar, hexagonal lattice of 42 by 36 formal neurons. Our aim in this analysis was to study in what context the "keys" (middle parts in the triplets) occur. The mapping of the $\mathbf{x}(t)$ vectors to the SOM was determined by the whole vector $\mathbf{x}(t)$, but after learning the map units were labeled according to the middle parts of the $\mathbf{m}_i(t)$. In other words, when the "key" parts of the different $\mathbf{m}_i(t)$ were compared with a particular word in the list of the selected 150 words (the most frequent ones), the map unit that gave the best match in this comparison was labeled by the said word. It may then also be conceivable that in such a study one should also use only such inputs $\mathbf{x}(t)$ for training that have one of the 150 selected words as the "key" part.

In order to equalize the mapping for the selected 150 words statistically and to speed up computation, a solution used in [2] was to average the contexts relating to a particular "key". In other words, if the input vector is expressed formally as $\mathbf{x} = [\mathbf{x}_i^T, \mathbf{x}_j^T, \mathbf{x}_k^T]^T$ where T signifies the transpose of a vector, and $\mathbf{x}_j$

is the "key" part, then the true inputs in the "accelerated" learning process were $[E\{\mathbf{x}_i^T|\mathbf{x}_j\}, 0.2\mathbf{x}_j^T, E\{\mathbf{x}_k^T|\mathbf{x}_j\}]^T$, where E now denotes the (computed) conditional average. (The factor 0.2 in front of $\mathbf{x}_j^T$ was used to balance the parts in the input vectors.) In this way there would only be 150 different input vectors that have to be recycled a sufficient number of times in the learning process. The information about all the 7624 words is anyway contained in the conditional averages. Although the above method already works reasonably well, a modification of "averaging" based on auxiliary SOMs was used. For each codebook vector a small, 2 by 2 SOM was assigned. It was trained with the input vectors made from the due word triplets. After training, each codebook vector in one small map described more specifically what context was used on the average with that "key" word.

The results of the computation are presented in Figure 27. The positions of the words on the map are solely based on the analysis of the contexts performed by the SOM. The general organization of the map reflects both syntactical and semantical categories. The most distinct large areas consist of verbs in the top third of the map, and nouns in the bottom right corner.

Word category maps can be used in practical large-scale natural language processing applications, like in intelligent information retrieval. This particular application area has been described in detail in the WEBSOM section of this report.

References

[1] Teuvo Kohonen. *Self-Organizing Maps.* Springer Series in Information Sciences, vol. 30. Springer, Heidelberg, 1995.

[2] H. Ritter and T. Kohonen. Self-organizing semantic maps. *Biol. Cyb.*, 61(4):241–254, 1989.

[3] T. Honkela, V. Pulkki, and T. Kohonen. Contextual relations of words in Grimm tales analyzed by self-organizing map. In F. Fogelman-Soulié and P. Gallinari, eds., *ICANN-95, Proc. of Int. Conf. on Artificial Neural Networks, Vol. 2*, pp. 3–7. EC2 et Cie, Paris, 1995.

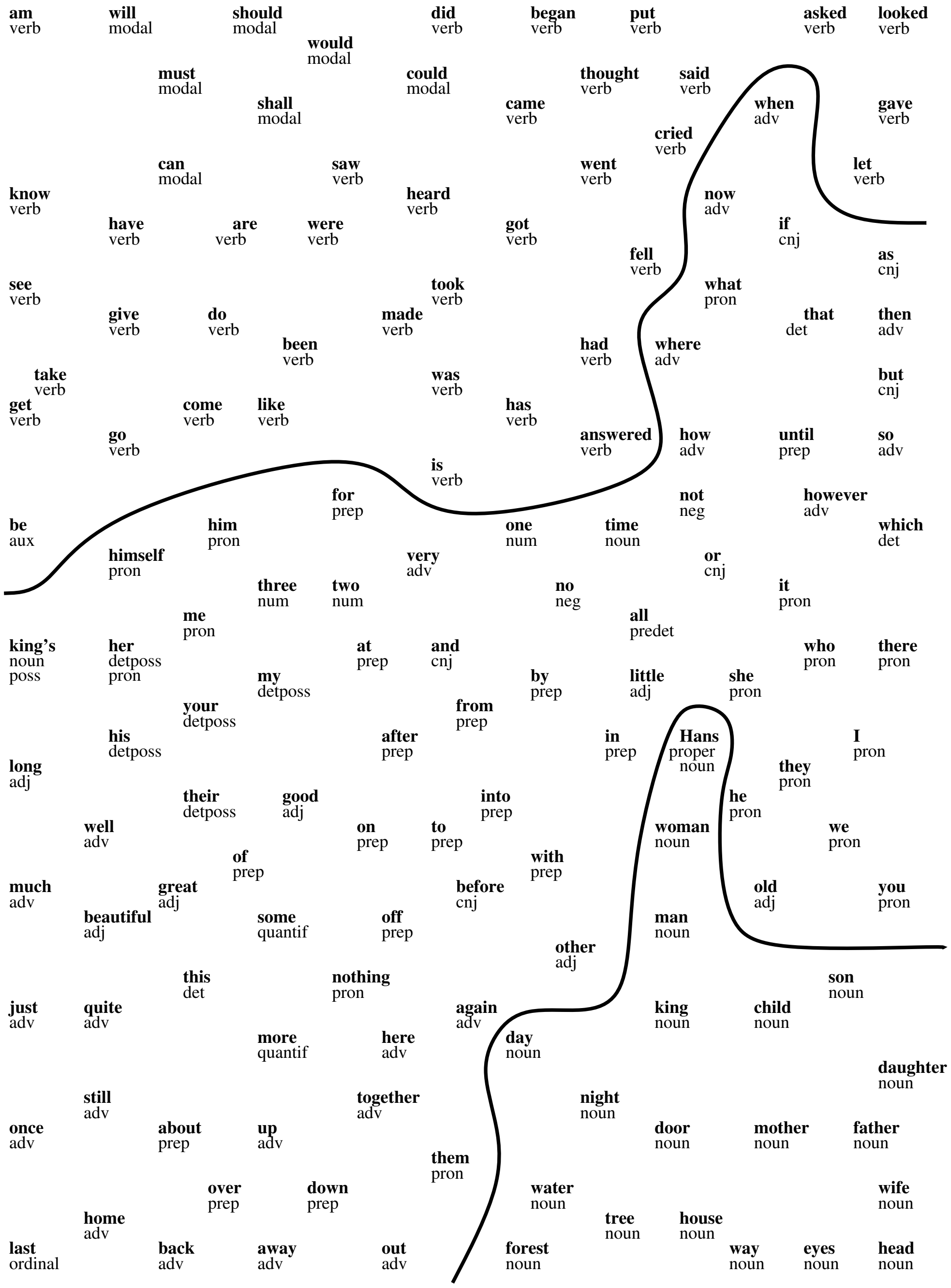


Figure 27: The 150 most frequent words of the Grimm tales, their statistical contextual relations being represented two-dimensionally by the SOM. The words are shown in their due position in the array; no symbols for "neurons" have been drawn. Many words are ambiguous but usually only the most common category relating to the tales is presented. All verbs can be found in the top section whereas the nouns are located in the lower right corner of the map. In the middle there are words of multiple categories: adverbs, pronouns, prepositions, conjunctions, etc. Modal verbs form a collection of their own among the verbs. Connected to the area of nouns are the pronouns. The three numerals in the material form a cluster. Among the verbs the past-tense forms are separated from the present-tense forms and located in the top right corner. Among the nouns, the inanimate and animate nouns forms separate areas of their own.

18 Speech Recognition

Mikko Kurimo and Panu Somervuo

18.1 The Recognition System

The recent projects in automatic speech recognition (ASR) are aimed both to use the recognition system as a test bench for the neural network algorithms developed in the laboratory and to develop the system itself as a pilot application of the neural networks. To produce respectable results, the best modeling and learning methods are applied with our own latest developments to fully exploit the available computer technology so that the recognition can still operate online in real time with high-dimensional input features. The results show that by the improved methodology and hardware, reductions in recognition error rate have been successful.

The speech recognition by the system developed in our laboratory occurs in five successive phases (see Figure 28). The most significant improvements have lately been introduced to the second and third phases. Some new inventions have also been tested for the spectrum analysis and for the phoneme string corrections.

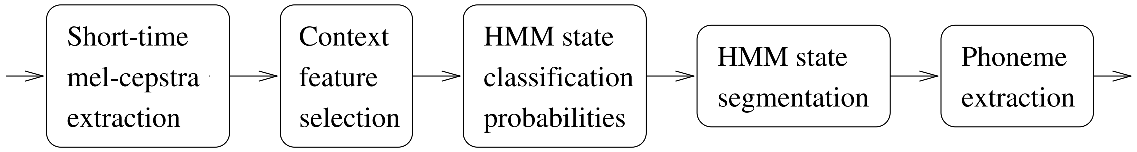


Figure 28: The main phases of the ASR by HMMs.

The recognition task used as a test bench for the new developments is the speaker dependent, but vocabulary independent ASR. The recognition is based on connecting the hidden Markov models (HMMs) of the phonemes to decode the phoneme sequences of the spoken utterances [5]. The HMM parameters can be automatically trained by neural network based methods using only a set of training words for each speaker. The output density function of each state in each model is a mixture of multivariate Gaussian densities.

18.2 Determination of the Error Rate

In the speech database collected here mostly in 1995, there are currently data of 20 speakers and at least four recording sessions of 350 Finnish words for each speaker. The speaker dependent recognition models are trained using three word sets and tested by the remaining one. The error rate given as the result is the number of all phoneme errors (inserted,deleted and changed phonemes) divided by the total number of phonemes. To gain statistical significance for the model comparisons, the tests are normally made for seven different speakers and the error rates are averaged. For verifying the robustness of the models for slightly different speech data also an older database (from 1990) is sometimes used. In general, the older database gives lower average error rates, probably because of the more experienced speakers.

For comparisons of the models the post-processing by the Dynamically Expanding Context (DEC) [1] is not applied in order to extract all the differences of the results. The long phonemes like /AA/ are separated from their short counterparts by using phoneme dependent duration limits learned iteratively during the model training. This simple separation do not take the word context into account and produces some errors which, in addition to some minor mismatches between the written and spoken format of the words, affect to the lowest obtainable value for the error rate. The acoustic features used throughout this work are the mel-cepstrum coefficients and RMS value of the signal. The basic feature vectors for the experiments are 20 component cepstra, but also extended feature vectors like averaged, concatenated and delta cepstra were tested and for those sometimes only 10–15 first coefficients were used.

18.3 Selection of Context Vectors and Multiple Feature Streams

The implementation of HMM is usually a first-order model and the context of short time feature vectors is thus not fully used. By using the context of the short-time feature vectors (see Figure 29), the coarticulation effect can be taken into account already in the feature extraction stage. The problem is how to define a suitable context. When the context is added to the recognition process there are two alternatives: whether to concatenate it to the short time feature vector or to use parallel feature streams in HMM so that the context is in its own feature stream.

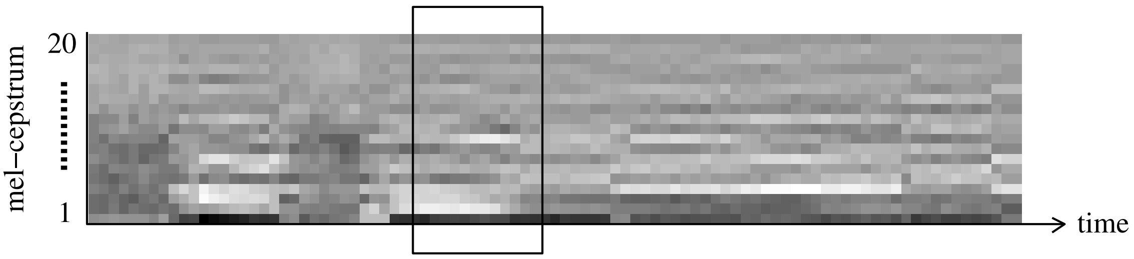


Figure 29: The features of the spoken word "OTANIEMI". The context vectors are combinations of several successive short-time features. The context window is 0.1s wide.

Several experiments were done in order to find discriminative features and a suitable context vector. Compared to the [4], new elements were delta features and the investigation of a proper time span for both static short-time feature concatenation and delta computations. One objective was to keep the dimension of the final feature vector suitably low. The effect of kernel width in SCHMM using Gaussian kernels was experimented with several feature vectors. All context vectors were found more tolerable to the change of kernel width than a single short-time feature. Compared to the case of using only one single mel-cepstrum, the use of three concatenated mel-cepstra dropped the phoneme recognition error from 6.7% to 3.5% and from 7.6% to 4.7% for two test speakers having different speaking rates. It was interesting that good recognition results were obtained even by using only delta features. Two

concatenated differences of mel-cepstra as a feature vector gave similar or even better results than one static mel-cepstrum vector. This is remarkable because when only difference features are used, the long time average of static features (channel bias) is automatically removed from the feature vector. The phoneme recognition errors for two test speakers using two concatenated mel-cepstra differences as a feature were 4.9% and 7.6%. When a static mel-cepstrum was concatenated to this feature vector, the corresponding errors were 3.1% and 6.0%. In general, significant improvements in recognition results were achieved when one or more additional static or difference mel-cepstra was concatenated to the single static or single difference mel-cepstrum. This shows the importance of using context of a single short-time feature vector.
Two alternatives, whether to concatenate the context vector into one feature stream or use parallel feature streams in HMM were experimented. When one static mel-cepstrum was used in one HMM input stream and two concatenated mel-cepstrum differences were used in another input stream, the phoneme recognition errors were 3.0% and 5.4% for test speakers. Here equal stream weightings were used. The latter error dropped to 5.0% when more weight was given to the static mel-cepstrum stream. The average phoneme error rate for six independent speakers using equal stream weighting was 5.6% when the feature selection was done according to the results obtained for two test speakers with different speaking rates. Compared to the baseline system, which had used only one static mel-cepstrum as a feature the phoneme error rate being 8.5%, the additional context stream gave the error reduction of 35%.
As a conclusion, the context of a single short-time feature vector is important and this study has shown that significant improvements in the recognition rate can be obtained by forming the context using only a few short-time feature vectors.

18.4 Scaling the Recognition System Up by using Extended Feature Vectors

Despite that the ASR systems should be able to operate online, it is also vital to study what will happen to the developed modeling and training method, when the dimensions are doubled or trebled. Actually, the computational capacity of the workstation used for the ASR demonstrations of the laboratory is now about five times than three years ago.
In the tests reported in Table 7 the dimensions are increased by adding delta features and concatenating averaged successive feature vectors into a high-dimensional context vector. The objective of these extended feature vectors is to provide the HMMs more freedom to create component-wise sequential dependencies by giving the observation densities of the states information on variable length features.
The recognition times in the Table 7 includes simple optimizations such as the partial distance computation and the ordered search mentioned in [3]for efficient computation with high-dimensional vectors. The HMMs in test were mixture density HMMs (MDHMMs) with 70 Gaussians per phoneme trained by SOMs and the segmental LVQ3. The demonstration system developed in 1994 with 24 Gaussians per phoneme gave the performance values 6.9 %, 9.8 % and 0.5 for the last three columns of the Table 7, respectively.
A completely different preprocessing approach has also been studied to develop

Feature vector	Cepstra	Cepstra Δ	RMS	RMS Δ	Context	Total dim.	Error rate% Data 90	Error rate% Data 95	Recognition time factor
basic	20		1		no	21	5.5	7.7	1.0
delta21	10	10	1		no	21	4.7	6.8	1.3
delta42	20	20	1	1	no	42	4.0	6.4	1.8
context80	15		1		5	80	3.6	5.3	1.7
context105	20		1		5	105	3.4	5.3	1.8

Table 7: The contents of the alternative feature vectors in the tests and the average test set error rates. The MDHMMs are trained by SOMs and the segmental LVQ3. The recognition time factor is the average recognition time per word divided by that of the baseline system ("basic" features).

feature extraction methods that correspond better the subjective voice observation of a human. By visual inspection the phonemes seem to be more distinct by the obtained auditive spectra than by the conventional mel-cepstra. One project is currently going on to test the auditive spectra for ASR.

18.5 Continuous Density Phoneme Models

The HMM structure has been a subject for a continuous development throughout the history of this work. The basic idea has been the simple temporal structure of uni-directional chains without skips (see Figure 30) and the principle of using one HMM for each of the 22 common Finnish phonemes including the silences directly before and after the word. For the output density of the states the building blocks have been Gaussians with a shared diagonal covariance matrix. The currently best performing version (Table 8) applies phoneme-wise tied Gaussian codebooks (PWMHMM), where the mixture densities are shared so that the states representing the same phoneme use the same codebook [2]. Thus there are as many sets of Gaussians as there are HMMs, which is a kind of intermediate for the traditional continuous HMMs (CDHMM) (different set for each state) and semi-continuous HMMs (SCHMM) (only one large set of Gaussians). The tied Gaussians resembles the vector quantization codebook of DHMMs, except that the densities are smoothly overlapped rather than partitioned.

By the PWMHMMs (and MDHMMs in general) the recognition occurs so that for the feature vector of every time window, the K-best matching Gaussians are extracted for every codebook and used to determine the HMM state classification probabilities. The codebooks were estimated from the training data by SOM and LVQ based training methods to ensure both the smoothness of the mapping and the efficient discrimination between phonemes. The most probable state segmentation for the feature sequence is then revealed using the HMM state structure and finally the phonemes are extracted from the path. When comparing the performance of the different continuous density HMMs the Table 8 shows that the PWMHMMs provide clearly the most appealing configurations, when the number of parameters, the recognition time and the error rate are compared.

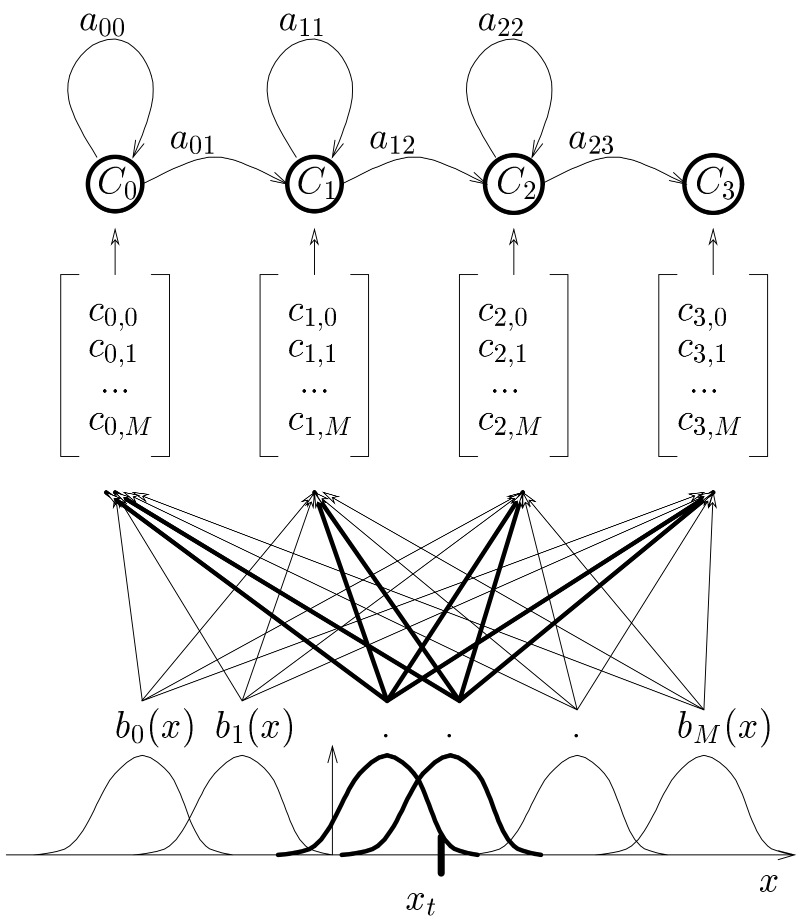


Figure 30: In phoneme-wise tied MDHMMs the same mixtures are used for the states representing the same phoneme. The model is defined by the set of transition probabilities a_{ij}, mixture weights $c_{i,k}$ and mixture densities $b_k(x)$. The output probability of state C_i at time t is approximated by using only the best matching mixture densities for the current observation vector x_t.

18.6 Discrete HMMs and Long Context Vectors using CNAPS

The experiments with high-dimensional context vectors [4] showed that the static recognition accuracy of separate phoneme tokens can be increased from 87% to 99% by substituting the single 20 dimensional cepstra by a 140 dimensional context vector and increasing the LVQ codebook size from 500 to 2000 units.
Since the winner search for the extended system was to slow to perform online in a normal workstation, the context vectors and the codebooks were sent to CNAPS parallel computer with 512 processing nodes. The search results are returned to the workstation for the HMM probability computations and the decoding of the most probable state sequence. To improve the HMM performance during the unstable transition parts between the phonemes, information from another LVQ codebook is also fed to the HMMs. The input for this codebook is the same as for the other codebook, but the classification task is to discriminate between phoneme centers and transition parts.

Type of HMM	Number of mixtures	Parameters (x10^3)		Recognition	
		weights	means	time factor	error%
CDHMM	4	0.4	9	0.7	7.9
PWMHMM	24	3	11	0.7	6.9
SCHMM	494	54	10	2.6	8.1
CDHMM	24	3	55	3.1	5.8
PWMHMM	70	8	32	2.1	5.7

Table 8: Some comparisons between different continuous density HMM structures. PWMHMM refers to the phoneme-wise tied mixture density HMMs. The CDHMM and SCHMM experiments are made using the same speech database (Data 90) and corresponding training methods. The recognition time factor is the average recognition time per word divided by that of the baseline system (24 mixture PWMHMM).

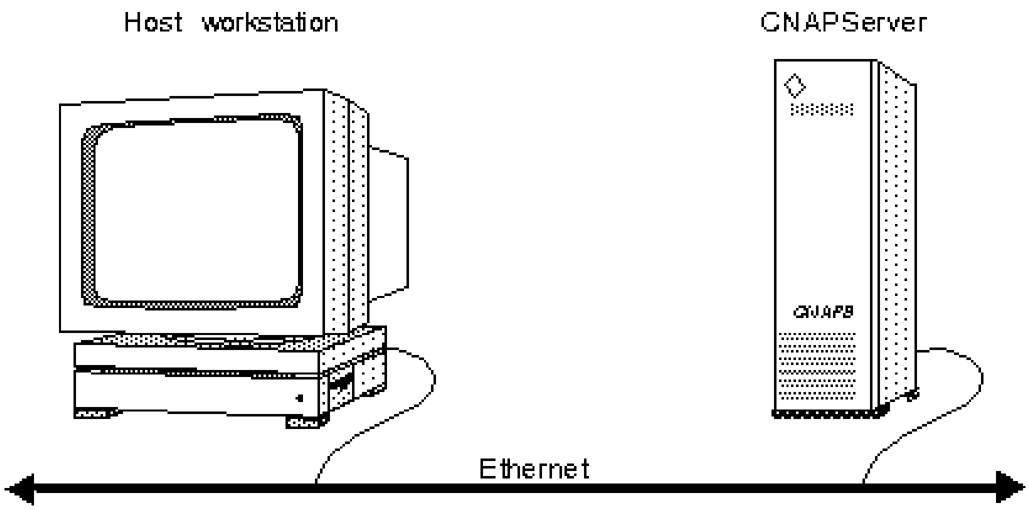


Figure 31: The CNAPServer System.

In the tested system the CNAPServer performed parallel computations under the control of a host workstation as shown in Figure 31. The context vectors are computed in the normal workstation and then transferred in buffers to the CNAPS. The information about the winner nodes is returned back to the workstation, which computes the HMM state classification probabilities and finally shows the decoded phoneme strings. However, despite the excellent off-line recognition results with DHMMs and the two large LVQ codebooks [4], the system working with the CNAPS had some problems in the fluent online operation.

18.7 CNAPS/PC Board in the Continuous Density System

While it was observed that the high-dimensional context vectors can also improve the performance of the new MDHMM system (see section 18.4), a CNAPS/PC board was tested to overcome the bottle neck occurring in the search of K-best match for all phoneme codebooks. The CNAPS/PC board is an ISA bus board that implements the CNAPS architecture for 128 processing nodes. The motivation for this work was also to see, whether there would be any major problems in transferring the workstation based system into the PC with a special board. At least the data communication was expected to be much simpler and faster than between the workstation and the CNAPServer.

The computations between the PC and the CNAPS are divided so that after the collection and formation of each context vector, the CNAPS immediately finds out the responses and the indexes of the K-best matching Gaussians. The PC then computes the state-dependent weighted sums that approximate the HMM state classification probabilities and takes care of the remaining phoneme decoding. So the CNAPS/PC board actually performs only a part of the third phase from whole the recognition process (see Figure 28), but this is the part that would otherwise take over 50% of total recognition time.

As a result of the study, a demonstration system operating with Linux PC using 80-dimensional context vectors processed by the CNAPS/PC board was able to perform the current ASR task smoothly online. The otherwise excessive codebook transfers were reduced by performing the parallel search on all phoneme codebooks in one operation, so that there is no need to change the codebooks in the processor memory. The size of the memory restricts, however, the use of larger codebooks and feature vectors. Due to the rapid capacity improvements in the general-purpose workstations, the corresponding ASR task is now processed online also in the 1997 recognition system without any special hardware.

18.8 Post-processing of Output Strings

If the set of possible output strings is known, the recognition error rate can be considerably reduced, even if the set is very large (e.g. 100 000 words). Successful post-processing can be applied as well for an open string set, if the correct strings are given corresponding to a set of evaluation samples.

A vocabulary-independent post-processing system for HMM based recognizers is shown in Figure 32. First it extracts the N best matching result strings using the mixture density hidden Markov models (HMMs) [3] trained by neural networks. Then the strings are corrected by the rules generated automatically by the Dynamically Expanding Context (DEC) [1]. Finally, the corrected string candidates and the extra alternatives proposed by the DEC are ranked according to the likelihood score of the best HMM path to generate those strings.

The objective of the system is to improve the HMM result strings so that the final result would be the best string allowed by the DEC rules. Since it is difficult to directly take care of the DEC rule base during the HMM decoding, the task is approached by transforming all the best HMM string candidates by the DEC. The ranking of the transformed strings is obtained by using another HMM decoding pass

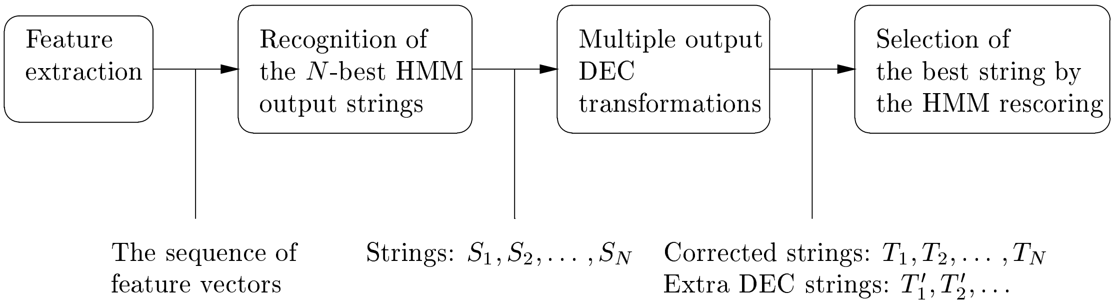


Figure 32: The stages of the N-best HMM-DEC decoding and the information that is transmitted between the stages.

which is this time restricted to the closed set of candidate strings. The experiments show that N need not be very large and the method can decrease recognition errors from a test data that even has no common words with the training data of the speech recognizer.

If the set of acceptable strings (the vocabulary) is available for the post-processing, a fast and efficient method based on Redundant Hash Addressing (RHA) [2] and two successive HMM decoding passes can be applied. First HMM decoding pass is made unconstrained and it provides one or more best-matching phoneme strings. Using these strings as keys for hashing, the large vocabulary can be quickly reduced by RHA so that only strings close enough to the keys will remain. The second, closed vocabulary HMM decoding pass can be then made in real time and a good approximation of the best matching acceptable string is found.

18.9 Other Activities

A separate report is given for the performance evaluation for a telephone based ASR application. Also the results from the developments of the neural network based training algorithms and fast density approximation methods are provided separately.

References

[1] T. Kohonen. Dynamically expanding context, with application to the correction of symbol strings in recognition of continuous speech. In *Proceedings of the 8th International Conference on Pattern Recognition*, pages 1148–1151, Paris, France, 1986.

[2] T. Kohonen and E. Reuhkala. A Very Fast Associative Method for the Recognition and Correction of Misspelt Words, Based on Redundant Hash-Addressing. In *Proceedings of the 4th International Conference on Pattern Recognition*, pages 807–809, Kyoto, Japan, 1978.

[3] M. Kurimo. Hybrid training method for tied mixture density hidden Markov models using Learning Vector Quantization and Viterbi estimation. In *Proceed-*

ings of the IEEE Workshop on Neural Networks for Signal Processing, pages 362–371, Ermioni, Greece, September 1994.

[4] M. Kurimo and P. Somervuo. Using the Self-Organizing Map to speed up the probability density estimation for speech recognition with mixture density HMMs. In *Proceedings of the International Conference on Spoken Language Processing*, volume 1, pages 358–361, Philadelphia, PA, USA, October 1996.

[5] J. Mäntysalo, K. Torkkola, and T. Kohonen. Mapping context dependent acoustic information into context independent form by LVQ. *Speech Communication*, 14(2):119–130, 1994.

[6] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

19 Using SOM and LVQ for HMM Training

Mikko Kurimo

19.1 New Training Methods for the HMMs

The training of the context-independent phoneme models for a minimal recognition error rate is difficult, because the variability of the phonemes in different conditions and contexts is substantial and the output densities of different phonemes do also overlap. A structure that can automatically adapt to all the complicated density functions, has a vast number of parameters and for proper estimation, the quality and quantity of the available training data is crucial. The size of the models and the training database demand robustness to the initial parameter values in order to avoid an excessively large number of training epochs and long training times.
The problem in practice with the widely spread training algorithms such as the segmental K-means (SKM) [8] and the segmental Generalized Probabilistic Descent (SGPD) [1] is that they sometimes converge slowly to low error rates unless good initial models are available.
Several common initialization methods have been compared for the mixture density hidden Markov models (MDHMM). The best results in terms of quickly obtained low final error rates in the automatic speech recognition (ASR) tests were obtained by using the Self-Organizing Maps (SOM) [2] to first train phoneme dependent codebooks and then use the codebook vectors as kernel centroids for the mixture densities. If the Learning Vector quantization (LVQ) [2] is used in the training after the SOMs, small improvements in the initialization can be achieved, but the SOM training can be performed much faster, because each phoneme codebook can be individually trained as a small SOM.

19.1.1 The Segmental SOM Training

The developed segmental SOM training for the HMMs [5] resembles to the conventional SKM type Viterbi training, but the main difference is that the parameters of mixtures belonging to the neighborhood of the best-matching component are also adapted. The motivation for the neighborhood adaptation is the parameter smoothing, where the level of the smoothing compared to the fitting accuracy to the training data is controlled by the neighborhood size. A wide neighborhood at the beginning ensures also that all the available codebook units will be drawn into useful regions in the input space. Compared to the codebooks trained without smoothing (e.g. by SKM) the accuracy provided by the best-matching Gaussian is usually worse, but that of the next $(K-1)$-best matches will be better, however, providing generalization for slightly discrepant characteristics of the test data.
The motivation to have ordered density codebooks is to enable accelerated state pdf estimation. In practice, a set of few best-matching kernels tend to dominate the density estimates for high-dimensional Gaussian mixtures, and thus the densities can be well approximated by excluding the other kernels. Since the search for the K-best matches consumes a significant part of the total computational load,

the search speed-ups have a significant effect on the total recognition speed. By exploiting the similarity of the successive feature vectors and the SOM topology in the mixtures, the approximate location of the K-best candidates can be determined accelerating significantly the state pdf estimation [5]. As the radius of the applied neighborhood function decreases gradually to zero the fine structure of the topology is lost due to the folding that increases the density estimation accuracy. However, some coarse structure will still be available to maintain smoothness and search acceleration capabilities (see Figure 33).

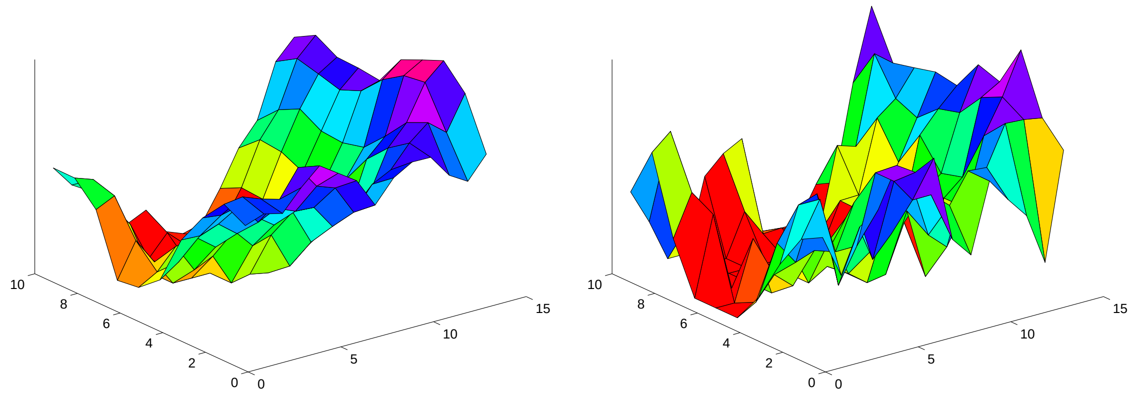


Figure 33: The responses of the individual mixture density components in the phoneme /A/ codebook organized into 10x14 grid are plotted for one randomly selected input vector. The first plot (left) is the situation when the radius is decreased to one and the second is after training with zero neighborhood

19.1.2 The Segmental LVQ3 Training and the LVQ2 Based Tuning

The segmental LVQ3 training [4]is in many ways similar to the segmental GPD improving the HMM parameters iteratively by comparing the best paths through the HMM states to the path producing the correct phoneme sequence for each training sample, updating the parameters and computing new paths. One of important difference is the lack of discrimination for situations, where the models already behave correctly in order to avoid extensive amount of adjustments to lower the state likelihoods. An other important difference is that the tuning is not directly dependent on the exact extent of the derivative of the whole word misclassification measure [1], but only on the relative difference of the modifiable parameter values to avoid the risk of improper learning step sizes for misses of variable error degree in one word.

The learning in the segmental training by both SOM and LVQ is here made in the batch mode, where each epoch includes the entire training data. The other possibility is to use a variable learning rate parameter to relate the modifications due to different training words. A proper definition of the learning rate would be difficult, however, because the parameter changes affect to the subsequent word segmentations. One method is, however, developed to train MDHMMs by LVQ that follows a pre-specified learning rate schedule between the training words. The method applies the LVQ2 type learning law to enhance the models by stochastic

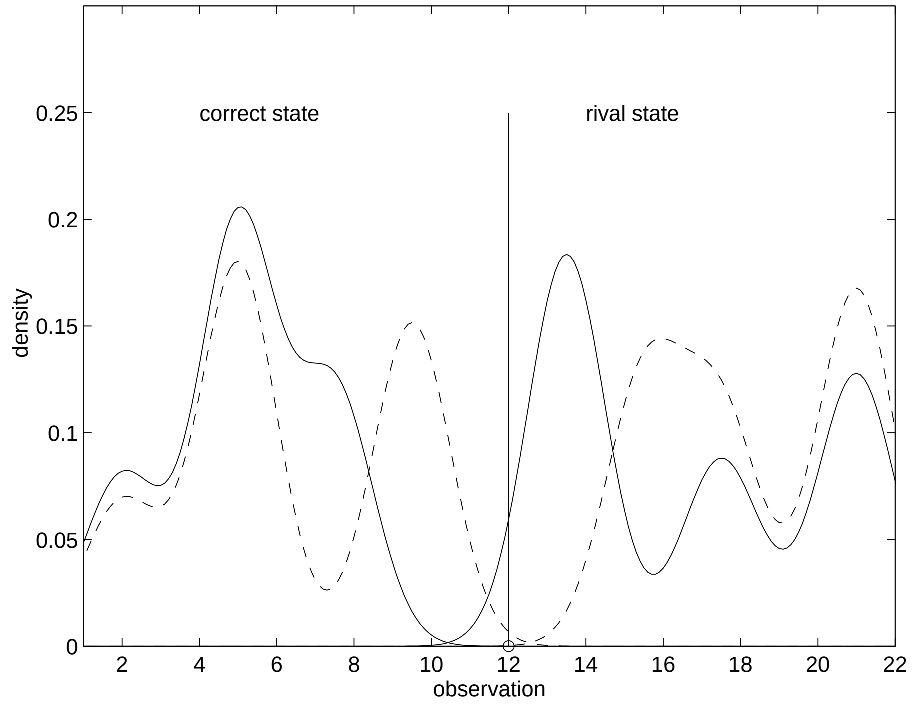


Figure 34: Adjusting the mixture densities of the competing states regarding to one observation (here, value 12). The parameters to be modified are the centroids of the nearest Gaussian for the correct state and, if a rival HMM state causes a misrecognition, also its corresponding centroid. The mixture weights of the modified mixtures are tuned respectively, but taking care of the normalization. The resulting new pdfs are shown dashed for this simple one-dimensional three-mixture case.

learning steps derived from the detected misrecognitions. This is suitable for a corrective fine tuning method, if the avoidance of the over-fitting in the training data can be controlled.

The criteria in the evaluation of the segmental training algorithms (see Part I of the Table 9) were the obtained average error rate for speakers on the both databases and that the low error rate level is achieved quickly even with initially inaccurate models. The suggested training in which the MDHMMs are first initialized by the SOMs and then trained by the segmental LVQ performed better than conventional methods with K-means initialization and SKM or SGPD training. By mixing the segmental training algorithms so that the models obtained by one is fed as an initialization to another, combinations can be found that eventually give lower error rates than the individual methods (Part II of the Table 9), but this requires much more training effort.

19.2 Increasing the Recognition Speed by Optimizing the Codebook Structure

When the dimension of the feature vectors and the size of the density codebooks are increased for better recognition accuracy, the bottle neck in online operation is the density approximation made by each HMM state for each feature vector in the

Initialization algorithm	HMM training algorithms	Error rate% Data 90 basic	Error rate% Data 95 context80
Part I			
KM	SKM	6.0	6.2
KM	SGPD	7.1	5.8
SOM	SLVQ3	5.6	5.3
Part II			
KM	SKM+SGPD	7.3	5.4
SOM	SLVQ3+SGPD	7.3	4.8
KM	SKM+LVQ2	5.5	5.6
SOM	SLVQ3+LVQ2	5.4	5.2

Table 9: Average test set error rates for alternative training methods after the initialization by K-means or SOM. The training methods are segmental K-means, segmental GPD, segmental LVQ3 and the corrective tuning based on LVQ2. In the Part I, the 5 epochs of HMM training is applied (no significant improvements was detected between 5 and 10 epochs). In the Part II, the last 5 training epochs were made with another algorithm (the improvement is significant, except for applying LVQ2 after the LVQ3) and the final error rates are given.

observation sequence. The topological K-best search was presented in [5] to give an example of a way to utilize the topology of an organized codebook for a fast approximative search algorithm for large codebooks. In addition of the topological order, this method assumes also that the successive feature vectors of speech usually resemble each other. Briefly, the search method presented in the Figure 35 begins by re-ranking the previous K-best matches and continues by checking the neighbors of the currently best match. If a new best is found, also the new neighbors are checked. This process continues until no more new best matches are found [6]. A complete search through the codebook is performed periodically to react for abrupt feature changes [7].
In the K-best search the fastest search time can be expected, if the candidates are ordered so that the most likely winners are checked first and the components of the feature vectors are processed in the order of decreasing significance. These characteristics are important, because each individual check of one candidate can be aborted immediately, when it becomes evident that it is not part of the K-best. With no special knowledge about the rank of the candidates except the continuous character of the signal, a good performance can be expected, if the candidates are scanned according to the distance in SOM topology from the expected winner. Similarly with no special knowledge about the rank of the components, it is best to orgarnize them according to the decreasing variance, in general.
The frequency of the complete search affects to the ability to react to fast changes in the signal characteristics and is, along with the number of the K-best matches and the size of the basic search neighborhood, a controllable variable to increase the accuracy or the speed of the search.

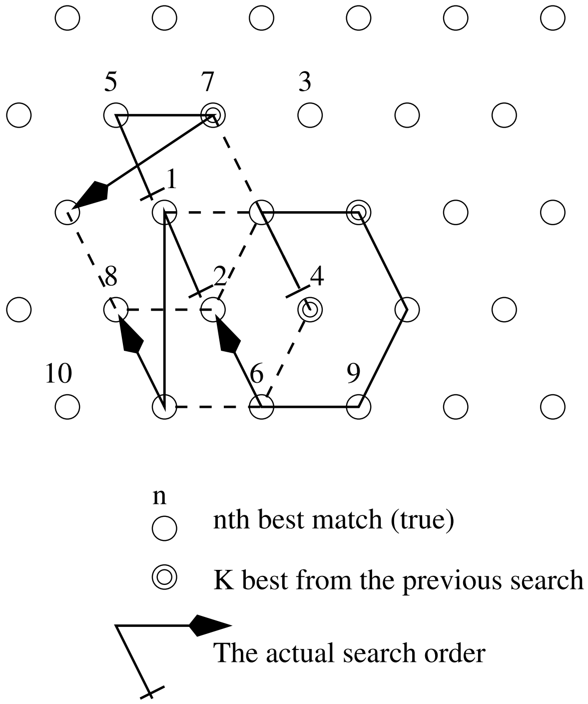


Figure 35: The topological search order for SOM codebooks.

The tree-search SOM [3] suits well to the fast approximative search for large codebooks, because the tree structure offers $O(\log N)$ search complexity instead of the normal $O(N)$. The possible loss of accuracy may follow from the sequential branching decisions by which most of the units are eliminated from the individual inspection. For K-best search the effect of the branching decisions is softened by expanding the search into the lower layer search areas associated with the rival best-matches from the upper layer. For density approximation purpose the Gaussian kernels are trained only for the lowest SOM layer and the upper layers act only as a search tool.

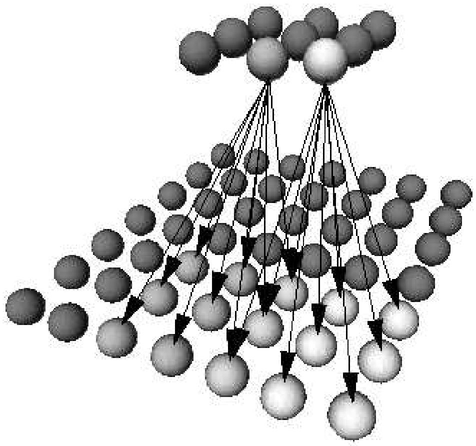


Figure 36: Two layers of the Tree-search SOM. In this map, each upper layer unit has a grid of 9 child units.

The results from the experiments indicate that the Tree-search SOM can be used as

a slightly worse performing, but a faster substitute to the normal SOM codebook. In the comparison by a ASR test to the corresponding normal SOM codebook, the Tree-search SOM, in which the recognition time decreased by 20%, increased the average number of recognition errors by 14%.
The topological K-best search compared to the unordered complete search offers a speed-up in the ASR experiments about 30–60% depending on the mixture sizes and feature vectors, while the increase of the average number of errors is only 4–10%. Despite the loss of most of the codebook topology, after the segmental LVQ3 training the same topological K-best search provide about 10% less recognition errors (about the same error rate as by complete search before the LVQ3). Thus, fortunately, the LVQ training seems to be more efficient to reduce the errors by increasing the discrimination than it is to generate them by destroying the topology required for fast search.

References

[1] W. Chou, B. Juang, and C. Lee. Segmental GPD training of HMM based speech recognizer. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 473–476, San Francisco,USA, april 1992.

[2] T. Kohonen. *Self-Organizing Maps.* Springer, Berlin, 1995.

[3] P. Koikkalainen and E. Oja. Self-organizing hierarchical feature maps. In *Proceedings of the International Joint Conference on Neural networks (IJCNN)*, volume II, pages 279–285, Piscataway, NJ, 1990. IEEE Service Center.

[4] M. Kurimo. Segmental LVQ3 training for phoneme-wise tied mixture density HMMs. In *European Signal Processing Conference*, volume 3, pages 1599–1602, Trieste, Italy, September 1996.

[5] M. Kurimo and P. Somervuo. Using the Self-Organizing Map to speed up the probability density estimation for speech recognition with mixture density HMMs. In *Proceedings of the International Conference on Spoken Language Processing*, volume 1, pages 358–361, Philadelphia, PA, USA, October 1996.

[6] J. Lampinen and E. Oja. Fast self-organizing by the probing algorithm. In *Proceedings of the International Joint Conference on Neural networks (IJCNN)*, volume II, pages 503–507, Piscataway, NJ, 1989. IEEE Service Center.

[7] E. Lopez-Gonzalo and L. A. Hernandez-Gomez. Fast vector quantization using neural maps for CELP at 2400 bps. In *Proceedings of 3rd European Conference on Speech Communication and Technology*, volume 1, pages 55–58, Berlin, Germany, September 1993.

[8] L. Rabiner, J. Wilpon, and B. Juang. A segmental K-means training procedure for connected word recognition. *AT&T Technical Journal*, 64:21–40, 1986.

20 Competing Hidden Markov Models on the Self-Organizing Map

Panu Somervuo

Models associated with the nodes of the Self-Organizing Map (SOM) can learn to become selective to the segments of temporal input sequences. Using the probability as a similarity measure between the input and the models leads to the concept of hidden Markov models (HMMs) as the nodes.
HMMs are stochastic signal models which have commonly been used in speech recognition. Their benefit is to tie separate observations in time together and utilize the time-dependency and order of acoustic phenomena in recognition while at the same time represent the speech patterns in a compact form as a state network. Besides speech recognition, HMMs have also been used in various other tasks, like natural handwriting recognition, text analysis, coding theory, ecology, and molecular biology.
Usually the training of the HMMs is supervised which requires that the segment units to be modeled are pre-defined. However, it might be advantageous to let the system choose the segment units itself. This was experimented in the present work. Input data may consist of unsegmented feature vector sequences with arbitrary lengths.
The unsupervised training of the segment models proceeds by utilizing the competitive-learning principles of the SOM. This is illustrated in Fig. 37.

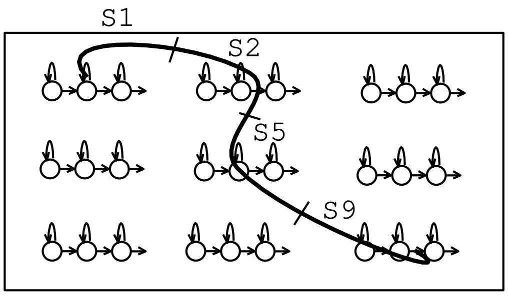


Figure 37: Competitive-learning of segment models on the SOM. Each map node is associated with an HMM having one or more states (three states in this case). The thick line represents the Viterbi segmentation of one input sequence. This corresponds to the best matching unit (BMU) search. The models of the BMUs and neighboring units are then updated by the corresponding segments.

Unsupervisedly derived segment models were experimented in the word recognition task. The recognition rate was 99.1% for a speaker-dependent system with the vocabulary of 350 Finnish words. This was equal to the best results of the supervisedly trained linguistic speech unit models.
The main result of this work was to demonstrate that the SOM gives a framework to train emergent state models by using unsupervised learning. A two-dimensional SOM array offers also a convenient way to visualize the state space of the recognition system.

21 Time Topology for the Self-Organizing Map

Panu Somervuo

In this work the time information of the input samples is taken into account when constructing the connections between SOM nodes. Those two nodes are connected which are the best-matching units for two consecutive input samples in time. This gives the time topology to the network. The reference models associated with the SOM nodes are first trained in the usual way, treating the input samples as static vectors and defining the neighborhood of the map nodes on the regular map grid. Once the map has been trained, old node connections are removed and new connections are created according to the time information of the input samples. The SOM training can then be continued by using the new node connections as a neighborhood when adapting the reference models. The result is a network where node connections represent temporal signal paths in the input space. Since any two nodes which are the best-matching units for two consecutive input samples in time can be connected independently of their Euclidean distance on the regular map lattice, the new connections may provide "worm-holes" to the original map lattice space.
In the following example, input data consist of sequences of two-dimensional feature vectors proceeding from the origo to the unit circle, see Fig. 38a. One-dimensional SOM with 100 nodes was constructed using this data. Figures 38b, 38c, and 38d illustrate three different map node connections when the reference vectors of the maps are kept the same. Fig. 38b shows the prototype vectors (depicted by dots) and the neighborhood connections (depicted by line segments) of the original one-dimensional SOM. Fig. 38c shows the connections created between the nodes which are the two best-matching units for each single input sample. Fig. 38d represents the time topology where the connections are created between the best-matching units of two successive input items in time. This gives a representation for temporal signal paths in the feature space. The network in Fig. 38d represents clearly best the original input data of Fig. 38a.

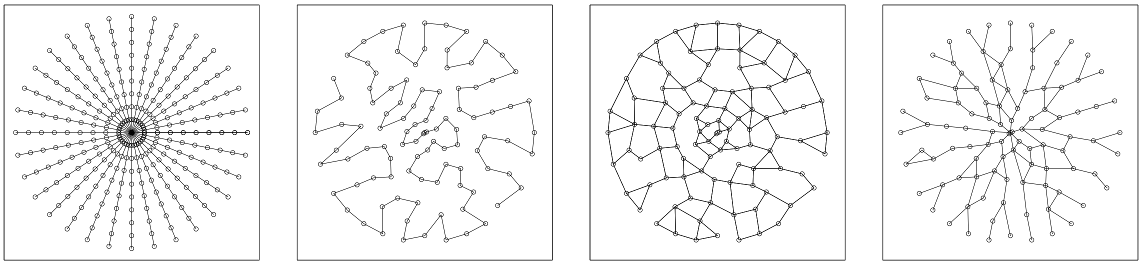


Figure 38: a b c d

The SOM with the time topology was experimented with speech data. 10-dimensional cepstrum vector sequences were computed from 1760 utterances of 20 speakers. Experiments consisted of training the SOM with the regular map lattice and with the time topology. The average quantization error and the word recognition error was then computed using a separate test set. The best results were achieved when the SOM was trained by using the time topology as a node neighborhood. Error in speaker-independent word recognition was 3.6 %.

22 The Self-Organizing Map and Learning Vector Quantization for Feature Sequences

Panu Somervuo and Teuvo Kohonen

The Self-Organizing Map (SOM) and Learning Vector Quantization (LVQ) algorithms are constructed in this work for variable-length and warped feature sequences. Instead of a single feature vector, an entire feature vector sequence is associated as a model with each SOM node. Dynamic time warping is used to obtain time-normalized distances between sequences with different lengths. In addition to the generalized median [2] (see also Sec. 5), an arithmetic average can be defined for feature vector sequences with different lengths [3]. Therefore both incremental learning and the Batch Map method can be used. Starting with random initialization, an ordered feature sequence map then ensues, and Learning Vector Quantization can be used to fine tune the prototype sequences for optimal class separation. The resulting SOM models, the prototype sequences, can then be used for the recognition as well as synthesis of patterns. Although time signals are here of main concern, warping can also be made in other dimensions. As pointed out in [1], many static processes can be reinterpreted as dynamic processes in which an artificial time coordinate is introduced.
Speaker-independent word recognition was experimented using one reference template for each of the 22 Finnish command words in the vocabulary. Recognition tests were repeated 20 times, each time having a different speaker in the test set and the remaining 19 speakers in the training set. 10-dimensional cepstrum vectors were used as features. The average recognition errors are given in Table 10.

reference templates	error, per cent
one randomly picked sequence from each class	18.5
one median sequence from each class	3.1
one LVQ-fine-tuned sequence for each class	1.5

Table 10: Speaker-independent word-recognition experiment with 1760 utterances from the vocabulary of 22 Finnish command words.

References

[1] R. Bellman, Dynamic Programming, Princeton University Press, Princeton, New Jersey, 1957; 6th printing 1972.

[2] T. Kohonen, "Self-organizing maps of symbol strings", Report A42, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland, 1996.

[3] D. Sankoff and J. Kruskal, Time warps, string edits, and macromolecules: the theory and practice of sequence comparison, Addison-Wesley, 1983.

23 Redundant Hash Addressing of Feature Sequences using the Self-Organizing Map

Panu Somervuo

Temporal sequences arise from various kinds of sources in the nature. Sensory elements transform the events into measurements and corresponding feature vectors. The present work addresses the question of how to efficiently process the feature sequences. Applications include retrieval, error correction, and recognition of sequential data. Due to the durational differences in the feature sequences and the variation and noise in the feature vectors, both temporal and spatial fluctuations must be tolerated in the sequence comparison. Dynamic programming (DP) based methods provide solutions for this, but they can be computationally heavy. A different approach is to use local fixed-sized features of the sequence. This facilitates the use of fast associative methods.
The present work combines two methods developed by Teuvo Kohonen. These are Redundant Hash Addressing (RHA) [1, 3] and the Self-Organizing Map (SOM) [3]. The central idea in the RHA is to extract multiple features from the same input item. The comparison of the input item against the reference items is based on these features. In case of character strings, segments of N consecutive letters (N-grams) have been used. The RHA system consists of the N-gram table and the dictionary, see Fig. 39. Multiple features (N-grams) are extracted from the input string and each extracted N-gram associates the input string with the dictionary items.

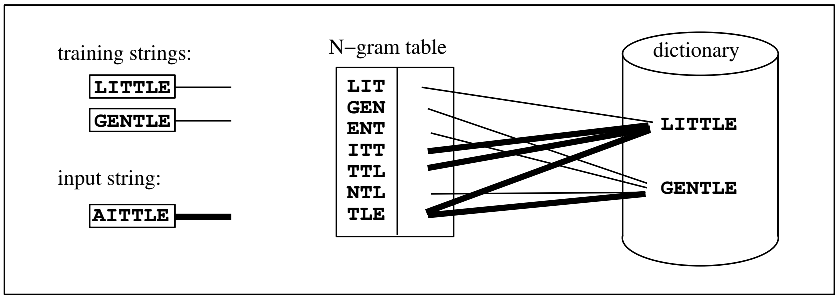


Figure 39: The RHA principle applied to character strings. The N-gram table is constructed by extracting the N-grams from training strings. Here trigrams are used ($N = 3$). From each item in the N-gram table there are associations (pointers) to the dictionary items. Associations activated by the erroneous input string 'AITTLE' are depicted by thick lines.

RHA has mainly been used for correcting textual output from speech recognizers, see e.g. [2]. In these, the recognition result is already in the form of a phoneme string. But in some applications, e.g. music processing, it is more difficult to extract, or even define, the underlying symbol sequence. Nevertheless, the RHA principle can still be used. As the RHA makes use of the N-grams of symbols and therefore the feature vectors must first be quantized, the SOM is used as a codebook to map the input feature vectors into the finite set of prototype vectors. When each SOM node is provided with an index, feature vector sequences can be mapped into symbolic

index sequences. Each feature vector is encoded by the index of its best-matching unit (BMU). The node indices of the SOM are thus the alphabet of the system.
Music retrieval and speech recognition experiments were carried out as a demonstration of the method. Mel-scaled spectrum vectors were computed from acoustic piano music samples. A 10-by-10-unit SOM was trained and the RHA table was constructed using 24 training sequences with duration of 10 seconds each. The test material consisted of 24 subsequences of the training sequences with the duration of one second. The beginnings of these subsequences were randomly chosen. As a result, for N=1, there were 2 erroneous retrievals, 4 completely correct retrievals, and 18 retrievals where the correct music piece shared a tie with an incorrect retrieval. For N=2, all retrievals were completely correct. Also for the values N=3, N=4, and N=5, all retrievals were completely correct.
The speech material used in the experiments consisted of 1760 utterances collected from 20 speakers (5 female speakers and 15 male speakers). The vocabulary was 22 Finnish command words. The recognition results are shown in Table 11.

feature	recognition method	error, per cent	time/ms
10-dim cepstrum	DTW $^{1)}$	3.4	78
BMU index	Levenshtein $^{2)}$	3.4	1060
BMU index	RHA N=2 $^{2)}$	6.6	5

Table 11: Multi-speaker speech recognition experiment. Averaged results of four independent runs. Time is the average recognition time for one input sequence. 1) one reference sequence per class, 2) 60 reference sequences per class.

Although the recognition accuracy of the RHA method is not as high as the accuracy using 10-dimensional cepstrum vectors and DTW, the recognition time is an order of a magnitude smaller. The BMU index sequences were matched against reference templates also by using the Levenshtein distance. This result shows that the recognition accuracy using BMU index sequences can be as good as using cepstrum vectors and DTW if multiple reference templates are used for each class, but this increases the recognition time considerably. A remarkable property of the RHA method is that adding new sequence templates to the recognition system does not slow its performance distinctly.

References

[1] T. Kohonen and E. Reuhkala, “A very fast associative method for the recognition and correction of misspelt words, based on redundant hash addressing”, in Proc. 4IJCPR, pp.807-809, Kyoto, Japan, Nov. 7-10, 1978.

[2] T. Kohonen, H. Riittinen, E. Reuhkala, and S. Haltsonen S, “On-line recognition of spoken words from a large vocabulary”, in INFORMATION SCIENCES, Vol. 33, pp. 3-30, 1984.

[3] T. Kohonen, Self-Organizing Maps, Springer Series in Information Sciences, Vol. 30. Springer, Heidelberg, 1995. Second ed. 1997.

24 Speech Recognition for the Hearing-Impaired

Panu Somervuo

Presently there are communication services available in Finland based on human speech-to-text (STT) and text-to-speech (TTS) interpretations, e.g. in telephone network and similar services in meetings attended by deaf persons. Such aid, based on human interpreters, is expensive and often problematic due to the intimate discussions that are interpreted.

The progress in speech technology has opened possibilities to automate these tasks. Finnish language appears to be very well suited to automatic STT conversion because the mapping from phonemes into graphemes is straightforward and the phonemic speech recognition has given promising results [2]. Therefore it could be possible to construct communication aids for the deaf and hard-of-hearing persons by using computer-based speech-to-text (STT) and text-to-speech (TTS) conversions.

By using a good phonemic speech recognizer in the STT conversion, the final word and content recognition could be left to the subject reading the raw output of a speech recognizer (grapheme string) on a screen. The other conversion direction, i.e., TTS synthesis, is no technical problem; several synthesizers exist for Finnish.

Recognition score requirements for STT conversion were assessed in the Laboratory of Acoustics and Audio Signal Processing at Helsinki University of Technology by simulating the reading of recognized messages. It was found that for isolated words the comprehension is good up to a 10 % phoneme error rate, for sentences up to 10-20 %, and for dialog sentences up to even 25 %. These results defined requirements for the recognition rate in the present application domain.

24.1 Experiments

The speech database was collected from 12 male speakers and 5 female speakers. The baseline speech recognition experiment was done with speech having 16 kHz sampling rate. A speaker-dependent speech recognizer [1] based on semi-continuous hidden Markov models was trained separately to each speaker. A 30-dimensional feature vector consisted of three concatenated mel-cepstra and the time window for its computation was 100 ms. Three first speech sets of each speaker were used in the training of the system and the fourth speech set was used for testing it. One speech set consisted of 350 Finnish words. Recognition error was computed as a number of inserted, deleted and changed phonemes in the recognized word divided by the number of the phonemes in the correct word spelling. The average phoneme recognition error was 9.1 % for male speakers and 8.9 % for female speakers. Another measure was computed as an amount of correct phonemes in the recognized word. These numbers were 93.2 % and 93.9 % for male and female speakers, respectively. Examination of the speech database revealed that there were missing endings and not well articulated words in some speakers' speech so that some of the phoneme errors using this speech database were due to the errors in the data.

For the evaluation of the speech recognition using analog telephone, the content of the database was filtered to the frequency range 300 Hz - 3400 Hz and downsampled to 8 kHz. The average phoneme errors were now 10.6 % for male speakers and

11.0 % for female speakers. The amount of correct phonemes was 92.0 %.
Some attempts towards the speaker-independent speech recognition were also made because this is a highly desirable feature in the target applications. As a by-product of this, new speaker clustering method was proposed. When the Self-Organizing Map is used as a codebook of each phoneme for each speaker, the similarity between two speakers can be defined as a distance between the phonemewise codebooks of the speakers. This allows the mapping of speakers into two-dimensional plane so that similar speakers are located near each other and speaker clusters can then be easily visualized.

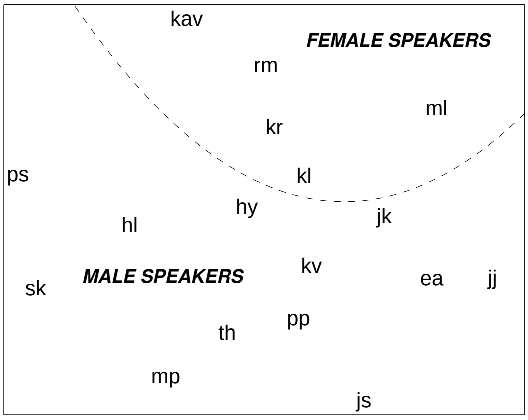


Figure 40: Speakers denoted by their initials are mapped into a two-dimensional plane so that similar speakers are located near each other. The similarity measure is based on the phonemewise Self-Organizing Maps of the speakers which form the basis of the speech recognition. The manually drawn dashed line shows that male and female speakers are discriminated.

The experiments reported here were done in the Neural Networks Research Centre as a feasibility study belonging to the project of the Laboratory of Acoustics and Audio Signal Processing at Helsinki University of Technology. The objective of this project was to experimentally investigate how different phonemic recognition schemes could be used in speech-to-text conversion aids for the hearing-impaired. Other set of experiments was done in the Speech and Audio Systems Laboratory, Nokia Research Centre. To our knowledge this was the first study where phonemic recognition was evaluated and shown to be a potential method for practical speech-to-text aids of the hearing-impaired.

References

[1] Kurimo, M. (1993) Using LVQ to enhance semi-continuous hidden Markov models for phonemes. Proc. of 3rd European Conf. on Speech Comm. and Tech., vol 3. pp. 1731-1734 20.

[2] Torkkola, K., Kohonen, T. et al. (1991) Status report of the Finnish phonetic typewriter project. In Kohonen, T. et al., editors, Artificial Neural Networks, volume 1, pp. 771-776. North-Holland.

25 Application of the Self-Organizing Map to the Categorization of Voice and Articulation Disorders, and to Exploration of Emotional Variation of Voice Quality

Lea Leinonen

A series of experiments was carried out to explore the applicability of the self-organizing map to visual imaging of voice and articulation disorders. The visual imaging technique was implemented on a portable PC for demonstrations. For acoustic categorization of voice qualities, a method was developed to select acoustic features with respect to their perceptual significance. The speech samples were assessed by experienced speech pathologists using auditory ratings along 6 dimensions: pathology, roughness, breathiness, strain, asthenia, and pitch.

The clinical studies suggested that visual imaging of speech with the self-organizing map could be used as a feed-back device during therapy: to show the deviation of voice quality from the norm, to show the deviation of phonemes from the norm (misarticulations of children, correction of articulation after cleft palate surgery), and to aid speech training of deaf children. In these applications visual feed-back is superior to auditory feed-back because even hearing subjects with voice or articulation disorders do not usually hear the difference between correct and incorrect performance. At present there is no such visual feed-back device for clinical use. Some training programs to support voice production and correct utterance of phonemes are commercially available for deaf children.

The self-organizing map, or the learning vector quantization, could also be used as diagnostic aid to measure: the degree and the quality of voice and articulation disorder before and after treatment, and deterioration of voice in provocation tests. At present, clinical evaluations are based on auditory ratings. The reliability of auditory rating tests is low because of high intra- and interrater variability. Repeated auditory ratings with several judges are difficult to carry out. For these reasons, the comparison of different surgical or other therapeutic maneouvres is difficult. The lack of reliable measures also restricts the diagnosis of voice disorders without visible anatomical changes, such as those induced by allergens or some inhaled medicines.

For all applications, statistically representative sets of speech data from healthy subjects and subjects with voice and articulation disorders are required. In our clinical studies speech samples were gathered from 200 subjects. This body of data proved to be too small for the selection of acoustic features for comprehensive clinical categorizations.

The self-organizing map was also applied to study emotional variation of voice quality. Spectral energy distributions, modeled by the map, showed differences among speech modes of anger, fear, asthonishment, sadness, scorn, plea, admire, and emotional neutrality.

References

[1] J. Kangas. On the analysis of pattern sequences by self- organizing maps. Doctor's thesis, Laboratory of Computer and Information Science, Helsinki University of Technology, 1994. Among other applications, some studies on pattern recognition of voice disorders were included in this thesis.

[2] L. Liu. Learning vector quantization and simulated annealing in automated feature selection for classification of voice disorders. Master's thesis, Department of Electronics, Helsinki University of Technology, 1995.

[3] H. Rihkanen, L. Leinonen, T. Hiltunen, and J. Kangas. Spectral pattern recognition of improved voice quality. *Journal of Voice*, 8:320-326, 1994.

[4] M.-L. Haapanen, L. Liu, T. Hiltunen, L. Leinonen, and J. Karhunen. Cul-de-sac hypernasality test with pattern recognition of LPC indices. *Folia Phoniatrica & Logopaedica*, 48:35-43, 1996.

[5] L. Leinonen, K. Valkealahti, and H. Rihkanen. Visual imaging of voice quality with the self-organizing map (in Finnish). *Suomen logopedis-foniatrinen aikaukauslehti*, 16:89-96, 1996.

[6] L. Leinonen and H. Poppius. Voice reactions to histamine inhalation in asthma. *Allergy*, 52:27-31, 1997.

[7] L. Leinonen, T. Hiltunen, M.-L. Laakso, H. Rihkanen, and H. Poppius. Categorization of voice disorders with six perceptual dimensions. *Folia Phoniatrica & Logopaedica*, 49:9-20, 1997.

[8] L. Leinonen, T. Hiltunen, I. Linnankoski, M.-L. Laakso. Expression of emotional-motivational connotations with a one-word utterance. *Journal of Acoustical Society of America*, 102:1853-1863, 1997.

26 Self-Organizing Map in Recognition of Topographic Patterns of EEG Spectra

Sirkka-Liisa Joutsiniemi and Samuel Kaski

Traditional automated EEG analysis methods detect abnormalities and suggest diagnoses on the basis of classifiers drawn from the EEG samples of different subject groups. The samples are collected from short artifact-free epochs that are chosen for the analysis by visual inspection; the evaluation of the whole record is practically not possible.

Methods that aim at distinguishing between different groups of samples require that there exist some pre-defined classes. The end result will be a separation of the classes based on a classifier that has been constructed using either a parametric model of the signals or a classifier that has learned to classify the available set of samples. Even neural classification methods have been applied to EEG signals; cf., e.g., [1-4]. When these kinds of methods are used, the whole work is concentrated on separating the classes, and no other information in the data samples than the class labels is considered important.

We aim at an EEG analysis method that would not need such predefined classes but which could *learn* representations of the different kinds of data types that there occur in the data set. The discovered data types can then be located on visual map displays, and these same data types can be detected quickly from new data samples by placing them on the same display. For example the time periods containing overwhelming muscle activity or eye blinks can be discarded from further analyses if necessary. What may be even more useful is that since no assumptions of the class structure of the data need to be made but instead the Self-Organizing Map tries to represent and illustrate the structures in the data, it may be possible to discover new structures that have not been apparent when the EEG signals have been visually inspected, as is traditionally done.

Our study was a pilot study where the goal was to verify the structures that the Self-Organizing Map discovers from multichannel EEG spectra. We used routine clinical EEG for which there exists a traditional classification that is predominantly based on the dominant frequency content of the signal, and that is correlated with the vigilance state of the subject. Also certain artifacts can be detected by a skilled EEG analyst. We used 6 classes in total, “a” for continuous alpha activity, “f” for flat EEG due to alpha attenuation, “t” for theta of drowsiness, “e” for eye movements, “m” for muscle activity, and “g” for bad electrode contacts. These classes were used in verifying the structures the Self-Organizing Map has revealed from the EEG. We investigated how well maps that had learned in a completely unsupervised manner were able to distinguish between these classes. After the capabilities of the SOM in EEG analysis have been verified in the pilot study, it is hoped that similar methods could be useful for both clinical routine monitoring of the EEG signal, and for searching for new structures and patterns from the signals.

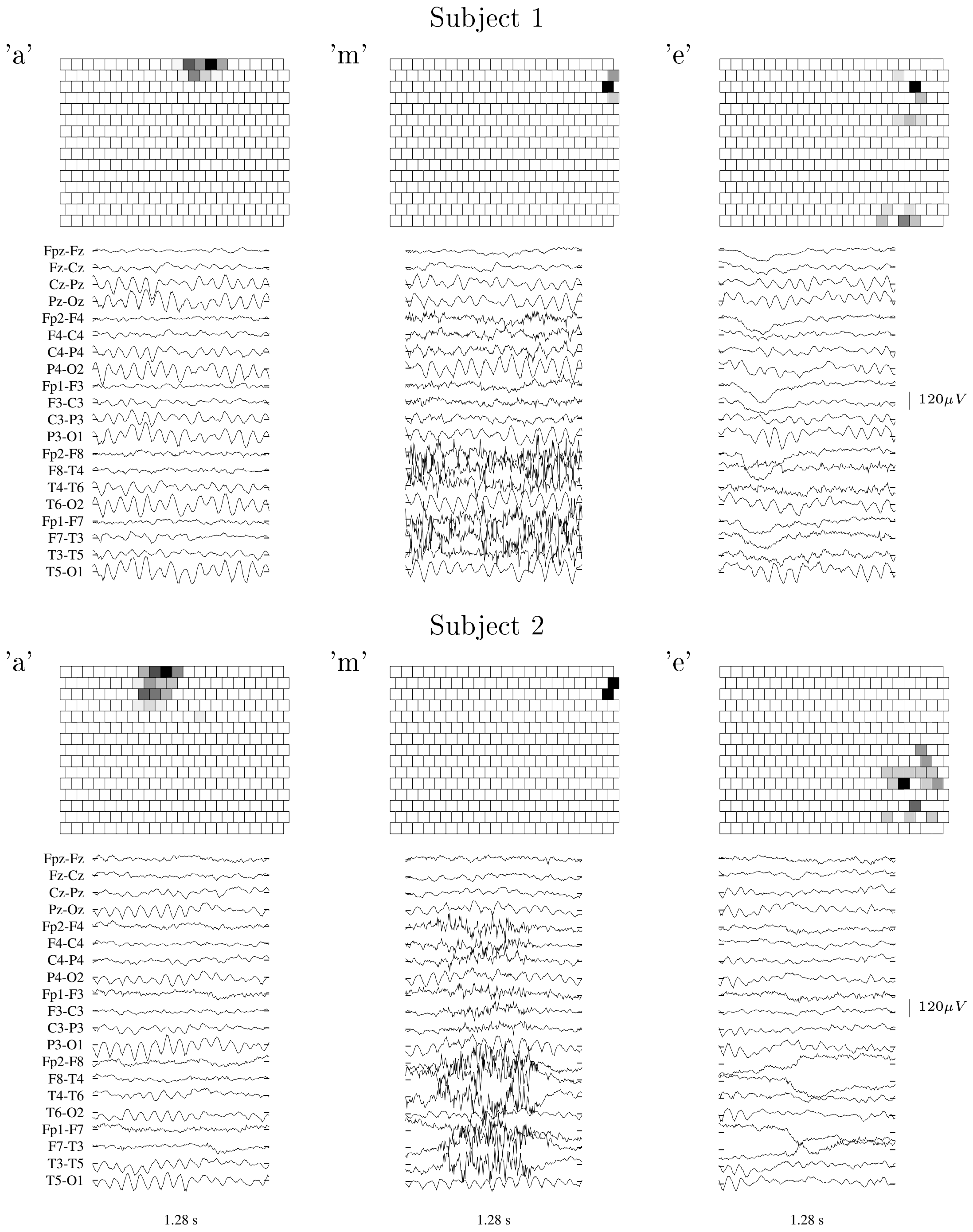


Figure 41: A self-organized map, taught with EEGs of 16 subjects, showing the locations of the EEG epochs with continuous alpha, “a,” with muscle activity, “m,” and with eye movements/blinks, “e,” in two subjects. In each small map, the squares stand for the 300 map locations. The shading indicates the projections of all labeled epochs in one recording, black color depicts the locations most often selected. An example of the epochs projecting onto the black units is shown below each map.

26.1 Methods

We used routine clinical 20-channel EEG signals recorded from 17 children. Short-time FFT (Fast Fourier Transform) was applied to each channel and the power in seven overlapping frequency bands was measured to generate the feature vectors that were input to the map. The features from each channel were used together, resulting in 140-dimensional vectors that describe the short-time frequency content all around the scalp.

26.2 Verification of the Results

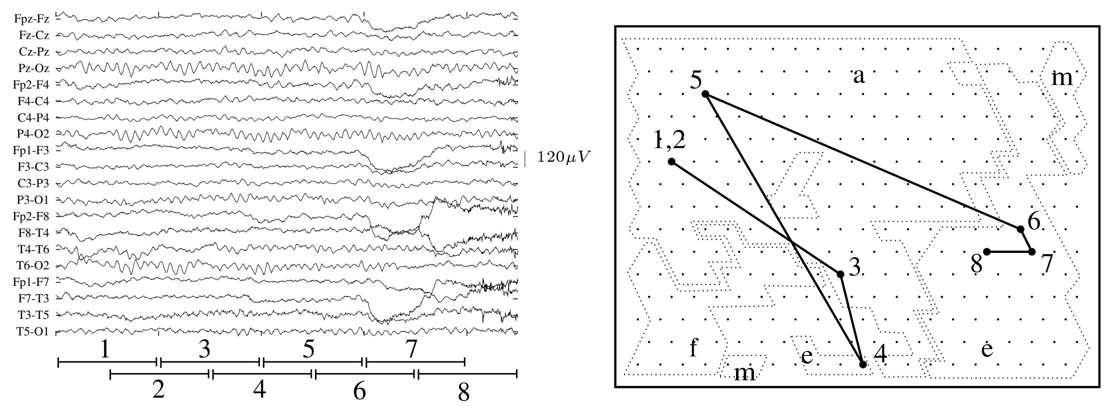
To verify the structures the map has extracted from the EEG data we tested how well the map was able to distinguish between certain types of EEG activity that are clearly discernible in the EEG signals even by naive analysts (examples are shown in Figure 41). The maps were able to correctly recognize these EEG-epochs chosen by an EEG analyst about 90% of the time; examples of the locations of the samples of different classes on the map have been shown in Figure 41.
The verification was made using subjects whose EEGs were not included in the teaching of the map, and the map learned in a completely unsupervised manner. The class information was only used for deciding which locations of the *already learned* map represent each of the classes.

26.3 Monitoring of EEG

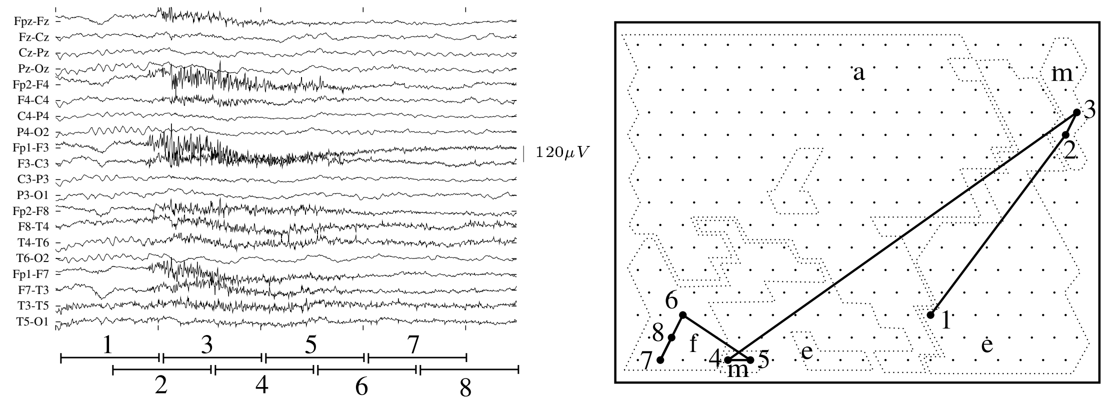
After the map has learned to represent EEG it can be used for monitoring of the EEG activity. Each EEG sample (a short-time multichannel EEG power spectrum) can be visualized as a point on the map, and successive samples form trajectories (Figure 42). Any auxiliary information can be used for creating labels on the map to aid in interpreting the trajectories. Unless the analyst has considerable amounts of experience in interpreting EEGs, the trajectories are much more easily interpretable than the set of original signals.

26.4 Discovery of Novel Patterns

If the same methods were applied to a set of EEG measurements collected in specially designed circumstances or from special groups of subjects, unexpected patterns might perhaps be recognizable either by inspecting the trajectories of the signals on the Self-Organizing Map, or the model vectors of the map. Most useful results would probably be obtained by tuning the feature extraction stage of the method to reflect any special nature of the experimental setting.
There exists an especially convenient method for visualizing the model vectors in the case of EEG signals: they can be plotted on an image of the scalp, in the location from which the measurements were made. A coarse display of this type is presented in Figure 43. The Self-Organizing Map display would be readily usable interactively: a click on a map location would result in the corresponding model vector to be visualized on an image of the scalp.



a



b

Figure 42: Locations of two continuous EEG segments of one subject on the map. **a** During the first segment, the subject is lying awake with his eyes closed. **b** During the second segment, the subject opens his eyes. The 1.28-s EEG epochs, each corresponding to a single location on the map, are indicated below the EEG records, and the locations of the epochs on the map are indicated by the numerals. The EEG segments were measured from a subject whose EEG was not used in the teaching of the map.

26.5 Summary

A Self-Organizing Map, taught with routine clinical EEGs of several subjects, was shown to be able to recognize similar topographic spectral patterns in different EEGs, also in EEGs not used for the teaching of the map. The results were verified by showing that the map differentiates between different types of background activities. The resulting map display can be used for both monitoring the ongoing EEG activity and for inspecting the types of activity there are in the individual EEGs.

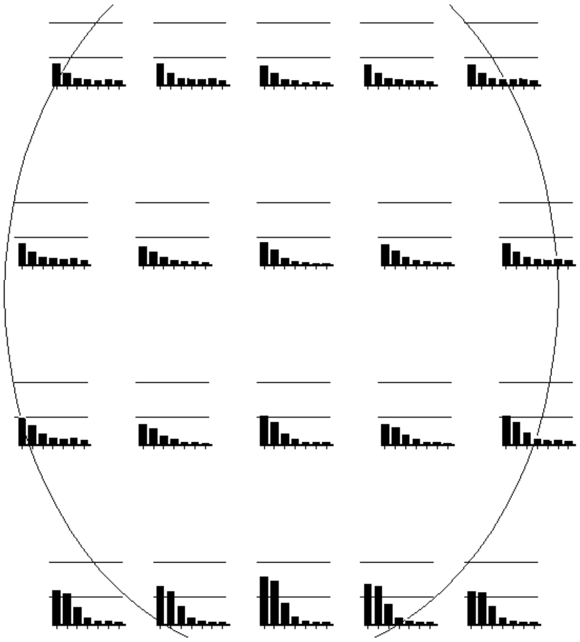


Figure 43: A model vector of an EEG map can be visualized as a topographic display of the feature values corresponding to each of the 20 channels. The small black bars give the amplitude of the feature components (activity at certain frequency bands on each channel). Here a model vector from the alpha, “a”, activity area of the map is shown.

References

[1] P. Elo, J. Saarinen, A. Värri, H. Nieminen, and K. Kaski. Classification of epileptic EEG by using Self-Organizing Maps. In I. Aleksander and J. Taylor, editors, *Artificial Neural Networks, 2. Proceedings of ICANN'92, International Conference on Artificial Neural Networks*, pages 1147–1150, Amsterdam, 1992. North-Holland.

[2] A. S. Gevins, R. K. Stone, and S. D. Ragsdale. Differentiating the effects of three benzodiazepines on non-REM sleep EEG spectra. a neural-network pattern classification analysis. *Neuropsychobiology*, 19:108–115, 1988.

[3] G. Jandó, R. M. Siegel, Z. Horváth, and G. Buzsáki. Pattern recognition of the electroencephalogram by artificial neural networks. *Electroencephalography and clinical Neurophysiology*, 86:100–109, 1993.

[4] G. Pfurtscheller, D. Flotzinger, W. Mohl, and M. Peltoranta. Prediction of the side of hand movements from single trial multi-channel EEG data using neural networks. *Electroencephalography and clinical Neurophysiology*, 82:313–315, 1992.

27 Increasing the Error Tolerance in Transmission of Vector Quantized Images by Self-Organizing Maps

Jari Kangas

In this study we consider an image compression system for its transmission error-tolerance properties. Efficient data compression is needed in image transmission applications. That is, because images consist of such a large number of elements. On the other hand, the image data can be compressed with high compression ratio, because the nearby pixels in images contain rather similar values.

The compression method used in this study is vector quantization (VQ), which is a powerful approach, but usually rather sensitive to transmission errors. It has been shown that the VQ codebooks produced by the Self-Organizing Map algorithm are comparable in quantization accuracy to codebooks designed by the LBG algorithm. Vector quantization is a very powerful compression method which naturally takes into account the redundancy of nearby image elements. The method considers vectors that are collected of nearby pixels in an image, a usual choice is to take a square of 4 by 4 (or 8 by 8) pixels. The compression happens when the vector is represented by an index to one model vector selected from a suitably designed codebook. The selection is done in such a way that the error occurring while replacing the original vector by the model vector is minimum.

The main problem in vector quantization is to design such codebooks that the average representation error over the compressed image is minimum. The above description is valid when there happens no errors in the index transmission. Then the codebook indexing can be done independent of the model vectors. If there are errors in the index transmission, we must consider the index relationships and the model properties together. The main problem is that if the codebook is indexed in an unordered manner, and if the transmitted index is changed to some other index, the representative model vector might change to a completely different model vector. The situation would be very different if we could order the codebook indexes in such a way that those codebook indexes which are easily mixed (due to transmission errors) would represent rather similar model vectors.

The Self-Organizing Map is trained in such a way that the model vectors in the map are spatially ordered after training, i.e., the neighboring model vectors in any place and in any direction of the Map are more similar than the more remote ones. This gives the idea of using the array coordinates of the Map as indexes in vector quantization. If we define the neighborhood function in the Self-Organizing Map training in such a way that the easily mixed models are always neighbors in the Map array, the training algorithm will ensure that the desired properties are achieved.

In other studies error-theoretic considerations were applied to the design of a VQ codebook for noisy environment. The algorithm turned out to be almost identical to the Self-Organizing Map algorithm. The identity was achieved when it was required that the training neighborhood was defined by the likelihood of changing an index to another one.

To demonstrate the performance of the SOM based error-tolerant image compression system, two transmission coding systems were designed. In the first scheme we used a digital pulse amplitude modulation (PAM) model with eight possible modulation amplitudes. The errors in the PAM model are amplitude level changes due to channel noise. As a second coding scheme we used a binary symmetric channel (BSC), where the errors were independent bit changes.

The SOM dimensions for codebooks had to be selected according to the principle of transmission. For the PAM transmission line, we selected a three- dimensional SOM, where there were 8 units in each dimension. The total number of codebook models is then 512. Each image block was transmitted over the PAM transmission line in three codes, one for each coordinate. Because the errors in each coordinate was independent of the others the probability of error in an index was considerably higher that the probability in a single code.

For the BSC channel we used a 9-dimensional SOM, where there were only two units in each dimension. The total number of units was then the same 512 as in the PAM model.

In the figures below two reconstructed images transmitted over a simulated PAM line are shown. In these the probability of errors was 0.1, which means that more than 40 % of the indexes were erroneous in the receiving end of the transmission line. In the image with random order the errors are usually rather severe. For example, in the middle of dark areas there are light blocks, and dark blocks are inserted in light areas. In the image with error-tolerant coding the errors are of different nature. For instance, in the dark area the erroneous blocks are never light, but "almost" dark. The subjectively experienced qualities of the images differ significantly, although the same number of errors were present in both.

Figure 44: The encoded and decoded images after transmission through a moderately noisy ($p = 0.01$) channel. The image on the left has been vector quantized with a nonordered codebook, and the image on the right with an ordered codebook, respectively. The subjectively experienced qualities of the images differ significantly, although the same *number* of codeword errors was present in both images.

28 Extraction of Features Using Sparse Coding

Harri Lappalainen

In various pattern recognition tasks the main problem is the extraction of features from raw data. Classification methods are well developed and they work generally very well if the features are suitable for the task. In some cases, physical or some other type of knowledge of the underlying structure of the patterns to be recognized guides the selection of the features. Usually, however, no such knowledge is available or it is incomplete. In these cases, the features have to be chosen heuristically, or by applying adaptive methods, which learn the features directly from the raw data. Sparse Coding combines many properties of two classical adaptive feature extraction methods: Principal Component Analysis and Vector Quantization.
The ability of sparse coding to extract features has been tested using natural images as raw data. A total of 15 images where used in the simulation. A part of one of the images is shown in figure 45. Figure 46 shows 16 out of 400 extracted filters. They are qualitatively similar to Gabor filters, which are widely recognized as a good set of features for image processing. This shows that sparse coding can be used to extract features from raw data.

Figure 45: A part of one of the natural images used in the simulation.

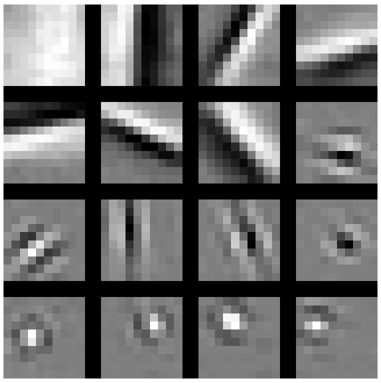


Figure 46: Some examples of features extracted from the images.

Research Projects in the Laboratory of Computer and Information Science

29 Neural Networks for Nonlinear Principal Component Analysis, Independent Component Analysis, and Blind Signal Separation

Erkki Oja, Juha Karhunen, Jyrki Joutsensalo, Aapo Hyvärinen, and Petteri Pajunen

During the past ten years or so, many neural network architectures and corresponding unsupervised learning rules have been introduced for performing standard Principal Component Analysis (PCA); for a review, see for example [4]. These world-wide developments are largely based on the pioneering work of the first two authors, initiated by E. Oja, in the beginning of the 1980's. We showed that relatively simple, neurobiologically justified Hebbian-type learning rules can provide PCA [1] and made a mathematical analysis of the related learning rules [2]. This early work was collected in the book [3]. Our work on neural PCA is now covered in many of the present-day textbooks on artificial neural networks. In fact, neural PCA is often seen as the other major paradigm in unsupervised neural learning, the other one being competitive learning and especially the Self-Organizing Map.
PCA networks have many applications in optimal linear representation of data in pattern recognition, data compression, and signal processing. They are especially suitable for on-line learning in situations, where it is not expedient to collect a data set and compute the PCA in batch mode. However, they have some inherent limitations, too, that have led researchers to study various forms of unsupervised neural learning beyond PCA. Such techniques are often collectively called nonlinear PCA methods. The main advantages of nonlinear PCA networks over standard PCA networks are:

1. The input-output mapping may be nonlinear while standard PCA is able to realize only linear mappings.

2. Higher than second-order statistics are taken into account in processing the input data via nonlinearities at least implicitly. This property is especially important in blind signal processing. Standard PCA is based on the use of covariances. These second-order statistics are sufficient for complete characterization of Gaussian data only, and for standard linear signal processing.

3. Neural realizations become more competitive compared to conventional numerical methods in nonlinear cases, because closed form solutions do not exist. Standard PCA can be determined efficiently using standard eigenvector computation routines.

Nonlinear PCA methods have applications in at least the following areas:

1. Robust PCA. Using suitable nonlinearities which grow less than linearly makes the analysis results more robust against outliers and non-Gaussian noise in the data. See Section "Robust fitting by nonlinear neural units".

2. Blind signal separation and Independent Component Analysis (ICA). See the corresponding section. These methods have applications in telecommunications, sensor array processing, medical signal processing, speech processing, financial time series analysis, and image feature extraction, to mention just a few of the most important application areas. We have applied methods developed in our laboratory to some of these problems with very interesting results; this will be discussed in more detail in later Sections.

3. Clustering of data and neural projection pursuit.

It is noteworthy that Nonlinear PCA methods, especially Independent Component Analysis, which is closely related to the blind signal separation problem, provide often a very meaningful representation of the input data. Furthermore, this representation is data dependent, and emerges in a completely unsupervised manner from the input data. Recently, Independent Component Analysis has been shown to be closely related to certain fundamental information-theoretic principles, such as maximization of output entropies of a neural network, minimization of mutual information, and information maximization. Currently, many leading neural network researchers share the opinion that these principles are fundamental in designing efficient neural network based information processing methods. Together with interesting applications, these facts have over the past few years prompted a great worldwide interest in neural realizations of Independent Component Analysis and related approaches.
We started a research project on ICA in 1994, based on our earlier theories of nonlinear PCA. Our research group is presently one of the leading ones in the world in this area, which is demonstrated for example by the many invited talks by Prof. Oja and Dr. Karhunen, invitations to international co-operation, visits etc.
In the next few sections of this report, sub-projects of the research effort in Nonlinear PCA and ICA neural networks will be covered in more detail by the members of the research group.

References

[1] E. Oja. A Simplified Neuron Model as a Principal Component Analyzer. *J. of Mathematical Biology*, vol. 16, 1982, pp. 267-273.

[2] E. Oja and J. Karhunen. On Stochastic Approximation of the Eigenvectors and Eigenvalues of the Expectation of a Random Matrix. *J. of Math. Analysis and Applications*, vol. 106, 1985, pp. 69-84.

[3] E. Oja. Subspace Methods of Pattern Recognition. *Research Studies Press and J. Wiley*, 1983.

[4] K. I. Diamantaras and S. Y. Kung. Principal Component Neural Networks: Theory and Applications. *J. Wiley*, 1996.

30 Nonlinear PCA Networks and Optimization Criteria

Juha Karhunen, Erkki Oja, Petteri Pajunen, Liuyue Wang, and Jyrki Joutsensalo

In this work, which was our earlier principal research topic in 1994 and before that, we have developed several different relatively simple nonlinear and robust generalizations of neural PCA methods.
A common rigorous approach to these developments is to derive new unsupervised neural learning algorithms by considering generalizations of the optimization criteria leading to the standard PCA solution. There exist several different optimization problems which lead to a standard PCA solution. These include:

1. Maximization of linearly transformed variances $\mathrm{E}\{[\mathbf{w}(i)^T\mathbf{x}]^2\}$ or outputs of a linear network under orthonormality constraints ($\mathbf{W}^T\mathbf{W} = \mathbf{I}$). Here $\mathbf{x}$ is the input (data) vector, $\mathbf{w}(i)$ is the weight vector of i-th neuron, and $\mathbf{W} = \mathbf{w}(1), \ldots, \mathbf{w}(M)$ is the weight matrix of a PCA network.

2. Minimization of the mean-square representation error $\mathrm{E}\{\| \mathbf{x} - \hat{\mathbf{x}} \|^2\}$, when the input data $\mathbf{x}$ are approximated using a lower dimensional linear subspace $\hat{\mathbf{x}} = \mathbf{W}\mathbf{W}^T\mathbf{x}$.

3. Uncorrelatedness of outputs $\mathbf{w}(i)^T\mathbf{x}$ of different neurons after an orthonormal transform ($\mathbf{W}^T\mathbf{W} = \mathbf{I}$).

4. Minimization of representation entropy.

In [2,3], we have derived a number of robust and nonlinear PCA learning algorithms from these generalized criteria for both symmetric and hierarchic network structures, and shown their relationships to existing neural PCA algorithms. In particular, generalization of the first variance maximization criterion leads for symmetric orthonormality constraint to the so-called Robust PCA rule:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu_k[\mathbf{I} - \mathbf{W}_k\mathbf{W}_k^T]\mathbf{x}_k\mathbf{g}(\mathbf{x}_k^T\mathbf{W}_k). \tag{81}$$

Here and later on the nonlinear odd function $g(t)$ is applied separately to each component of its argument vector. The index k denotes iteration or sample number, and μ_k is the learning parameter at iteration k. This rule has been shown to be useful in clustering, projection pursuit, and robust PCA. It is often useful to preprocess the data vectors $\mathbf{x}_k$ by whitening (sphering) them. After this, the learning rule (81) responds directly to higher-order statistics in the data.
Similarly, generalization of the second optimization problem, minimization of the mean-square representation error, leads to so-called Nonlinear PCA rule:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu_k[\mathbf{x}_k - \mathbf{W}_k\mathbf{g}(\mathbf{y}_k)]\mathbf{g}(\mathbf{y}_k^T), \tag{82}$$

where the output vector $\mathbf{y}_k = \mathbf{W}_k^T\mathbf{x}_k$. We have shown in several papers, summarized in [4], that with prewhitening the learning algorithm (82) can be successfully applied to blind separation of certain type source signals. The blind separation problem is discussed in several other sections of this report. The nonlinear PCA rule provides an especially simple neural solution to this difficult problem. This has been analyzed rigorously in [4,10].

The learning rules (81) and (82) were proposed on intuitive grounds already in [7]. Later on, their relationship to optimization problems were made rigorous in the theoretical papers [2,3]. We have also developed a number of other algorithms using this optimization based approach to nonlinear PCA; see [2,3,6]. In particular, the so-called bigradient algorithm developed and analyzed in [6] provides a versatile tool. In various forms, it can be applied both to robust PCA problems, making the results insensitive to outliers in the data and inpulsive noise, as well as to blind source separation.

We have also developed fast converging approximative least-squares algorithms [5] for minimizing so-called nonlinear PCA criterion given by

$$J(\mathbf{W}) = \parallel \mathbf{x} - \mathbf{W}\mathbf{g}(\mathbf{x}^T\mathbf{W}) \parallel^2 \tag{83}$$

These least-squares algorithms can again be applied to blind separation of sources after prewhitening of the input data [5]. - The same criterion (83) is used as a starting point in deriving the Nonlinear PCA rule (82), too.

Recently, we have derived new results on the nonlinear PCA criterion (83) in blind source separation and related problems [8,9]. The criterion can be expressed for prewhitened data in a simple form. This allows an easy comparison with other criteria used in blind signal processing and independent component analysis, including cumulants, Bussgang criteria, and information theoretic contrast functions. The results show the close connection of the nonlinear PCA learning rule (82) with certain well-known other algorithms used for blind source separation, and help in the optimal choice of the nonlinearity [8,9].

Still other theoretical results include stability considerations of these algorithms. In [1], a rigorous stability condition has been derived for PCA subspace rule, and the stability of the robust algorithm (81) is shown to be better if the the nonlinear function $g(t)$ grows less than linearly.

References

[1] J. Karhunen. Stability of Oja's PCA subspace rule. *Neural Computation*, 6:739–747, 1994.

[2] J. Karhunen and J. Joutsensalo. Representation and separation of signals using nonlinear PCA type learning. *Neural Networks*, 7(1):113–127, 1994.

[3] J. Karhunen and J. Joutsensalo. Generalizations of principal component analysis, optimization problems, and neural networks. *Neural Networks*, 8(4):549–562, 1995.

[4] J. Karhunen, E. Oja, L. Wang, R. Vigario, and J. Joutsensalo. A class of neural networks for independent component analysis. *IEEE Trans. on Neural Networks*, 8:486–514, 1997.

[5] P. Pajunen and J. Karhunen. Least-squares methods for blind source separation based on nonlinear PCA. *Int. J. of Neural Systems*, 8(5-6):601–612, 1997.

[6] L. Wang and J. Karhunen. A unified neural bigradient algorithm for robust PCA and MCA. *Int. J. of Neural Systems*, 7(1):53–67, 1996.

[7] E. Oja, H. Ogawa, and J. Wangviwattana. Learning in nonlinear constrained Hebbian networks. In T. Kohonen et al. (Eds.), *Artificial Neural Networks* (Proc. ICANN'91, Espoo, Finland, June 1991). North-Holland, Amsterdam, 1991, pp. 385-390.

[8] J. Karhunen, P. Pajunen, and E. Oja.. The nonlinear PCA criterion in blind source separation: relations with other approaches. *Neurocomputing*, 22:5–20, 1998.

[9] E. Oja. Nonlinear PCA criterion and maximum likelihood in independent component analysis. In *Proc. First Int. Workshop on Independent Component Analysis and Signal Separation (ICA'99).* Aussois, France, January 1999, pp. 143-148.

[10] E. Oja. The nonlinear PCA learning rule in independent component analysis.. *Neurocomputing*, 17:25–45, 1997.

31 Blind Signal Separation and Independent Component Analysis

Erkki Oja, Juha Karhunen, Jyrki Joutsensalo, Aapo Hyvärinen, Petteri Pajunen, and Ricardo Vigário

Both Principal Component Analysis (PCA) and Independent Component Analysis (ICA) [2] attempt to find a coordinate transformation of a collection of multivariate data, by which the new coordinates or feature dimensions have some desirable properties in terms of data compression and representation. In the case of classical PCA, the new coordinates are uncorrelated and an optimal linear compression is achieved in the minimum mean square sense. In the case of ICA, the new coordinates are statistically independent, which means that a very efficient data representation is possible.
An especially interesting connection of ICA exists to the problem of Blind Signal Separation [1,3]. A mathematical definition is the following: an L-dimensional vector-valued discrete signal $\mathbf{x}_k = [x_k(1), \ldots, x_k(L)]^T$ at the discrete time k is assumed to be of the form

$$\mathbf{x}_k = \mathbf{A}\mathbf{s}_k + \mathbf{n}_k = \sum_{i=1}^{M} s_k(i)\mathbf{a}(i) + \mathbf{n}_k. \tag{84}$$

Here $\mathbf{s}_k = [s_k(1), \ldots, s_k(M)]^T$ is a source vector consisting of M *unknown* source signals (independent components) $s_k(i)$ $(i = 1, \ldots, M)$ at time k. $\mathbf{A} = [\mathbf{a}(1), \ldots, \mathbf{a}(M)]$ is a fixed $L \times M$ *unknown* mixing matrix whose columns $\mathbf{a}(i)$ are the basis vectors of ICA, and $\mathbf{n}_k$ denotes possible corrupting additive noise. The noise term $\mathbf{n}_k$ is often omitted from (84), because it is usually impossible to distinguish it from the source signals. Instead of time, k can also stand for the spatial location of a pixel, like in the example of Figs. 47, 48.
The problem is to find the mixing matrix $\mathbf{A}$, when only a sample $\mathbf{x}_k, k = 1, 2, ...$ of the mixtures is available.
The following assumptions are typically made [1]:

1. $\mathbf{A}$ is a constant matrix with full column rank. Thus the number of mixtures L is at least as large as the number of sources M, which is usually assumed to be known in advance. If $M < L$, the data vectors $\mathbf{x}_k$ roughly lie in the M-dimensional subspace spanned by the basis vectors of ICA.

2. The source signals $s_k(i)$ $(i = 1, \ldots, M)$ must be *mutually statistically independent* at each time instant k, or as independent as possible. The degree of independence can be measured using suitable constrast functions.

3. Each source signal $s_k(i)$ is a stationary zero-mean stochastic process. Only one of the source signals $s_k(i)$ is allowed to have a Gaussian marginal distribution.

Note that very little prior information is available for the matrix $\mathbf{A}$. Therefore, the strong independence assumptions are required to fix the ICA expansion (84). Even then, only the directions of the ICA basis vectors $\mathbf{a}(i)$, $i = 1, \ldots, M$, are defined.

To get a more unique solution, one can normalize the variances of the source signals to unity.
In the techique called *blind source (or signal) separation*, one tries to extract the unknown waveforms $\{s_k(i)\}$, $k = 1, \dots$, of the independent source signals in (84) from the data vectors $\mathbf{x}_k$ by a linear transformation

$$\mathbf{y}_k = \mathbf{B}\mathbf{x}_k, \tag{85}$$

where $\mathbf{B}$ is called a separating matrix. The elements of $\mathbf{y}_k$ approximate the source signals $s_k(i)$. Such blind techniques are useful for example in array processing, speech enhancement, and communications. A typical example is the "cocktail party effect": suppose we can record the mixed voices from a party by several microphones. The blind source separation would give the voices of the individual speakers.
In several blind separation algorithms, the data vectors $\mathbf{x}_k$ are preprocessed by whitening (sphering) them, so that their covariance matrix becomes the unit matrix. After whitening, the separating matrix $\mathbf{B}$ can be assumed orthogonal. This auxiliary constraint makes the separating algorithms simpler, and also normalizes the variances of the estimated sources automatically to unity.
A practical difficulty in designing source separation and ICA algorithms is reliable verification of the independence condition. It is impossible to do this directly because the involved probability densities are unknown. Therefore, approximating contrast functions which are maximized by separating matrices have been introduced [2]. As an example, for prewhitened input vectors it can be shown that the relatively simple contrast function based on the fourth-order cumulant or *kurtosis*

$$J_1(\mathbf{y}) = \sum_{i=1}^{M} \mid \mathrm{cum}[y(i)^4] \mid \; = \; \sum_{i=1}^{M} \mid [\mathrm{E}\{y(i)^4\} - 3\mathrm{E}^2\{y(i)^2\} \mid \tag{86}$$

is maximized by the separating matrix $\mathbf{B}$ in model (85), if the sign of the (unnormalized) kurtosis $\mathrm{cum}[s(i)^4]$ is the same for all the source signals $s_k(i)$, $i = 1, \dots , M$.
A 3-layer feedforward network was proposed in [4] for ICA and blind source separation. Each of the 3 layers performs one of the processing tasks required for complete ICA: 1. whitening; 2. separation; and 3. estimation of the mixing matrix. Any of these three tasks can be performed either neurally or conventionally.
For whitening, simplified versions of neural PCA learning rules are convenient. For separation, we can use the *nonlinear PCA rule* [5]:

$$\mathbf{W}_{k+1} = \; \mathbf{W}_k + \mu_k[\mathbf{x}_k - \mathbf{W}_k\mathbf{g}(\mathbf{y}_k)]\mathbf{g}(\mathbf{y}_k^T). \tag{87}$$

with μ_k the learning rate, $\mathbf{x}_k$ the mixture vectors that are now assumed whitened, and $\mathbf{y}_k = \mathbf{W}_k^T\mathbf{x}_k$ the output vector from a neural layer whose weights are given by matrix $\mathbf{W}_k$. The function $g(.)$ is a suitable nonlinearity, e.g. the hyperbolic tangent function. During learning, the weight matrix $\mathbf{W}_k$ converges to a (transposed) separating matrix [5], and the elements of $\mathbf{y}_k$, or the outputs from the neural layer, tend to the source signals.
A connection of nonlinear PCA to some other statistical and information theoretic criteria, as well as the learning rules, are discussed in another Section of this report.
In 1995, we also developed another so-called *bigradient algorithm* [6], which is applied for learning the orthonormal separating matrix $\mathbf{B}$ as follows:

$$\mathbf{W}_{k+1} = \; \mathbf{W}_k + \mu_k\mathbf{x}_k\mathbf{g}(\mathbf{y}_k^T) + \gamma_k\mathbf{W}_k(\mathbf{I} - \mathbf{W}_k^T\mathbf{W}_k). \tag{88}$$

Here γ_k is another gain parameter. usually about 0.5 or 1 in practice. Again, the weight matrix $\mathbf{W}_k^T$ tends to the separating matrix $\mathbf{B}$.
Since 1996, new algorithmic development into the ICA and BSS problem has concentrated on the fixed-point learning rules, implemented in the FastICA software package (see the section on one-unit and fixed point ICA algorithms). Also several extensions have been studied recently, like nonlinear mixing models, robust algorithms, and relations with complexity criteria - see the separate section on extensions.
Our ICA / BSS ideas have been applied to a number of artificial and real signals, e.g. to separate 10 speech signals from their mixtures. As an illustrative example, Fig. 47 shows 9 mixtures of 9 natural images. This means that the 9 original images (not shown) have been multiplied pixel-wise by randomly chosen coefficients and added together, to obtain one of the mixtures shown here. Different multiplying coefficients have been used for the 9 different mixtures. The 9-dimensional mixture vectors $\mathbf{x}_k$ in eq. (84) are obtained by collecting the gray levels of pixels in the 9 mixture images at the same pixel location. Thus k is a running index for the pixel location. In this experiment, there was no additive noise in the mixtures. These mixture vectors where whitened by PCA and input to the nonlinear PCA learning rule, eq. (87). The outputs after learning, again collected into images, are shown in Fig. 48. These are quite close to the original images used in forming the mixtures. Note that no information whatsoever was used on the mixing coefficients (elements of matrix $\mathbf{A}$) or the original images in computing these results. The only information the algorithm had were the mixtures of Fig. 47.
More conrete applications are in biomedical signal analysis, financial time series analysis, and feature extraction for digital images. All of these are covered in their separate Sections in this report.

References

[1] J.-F. Cardoso and B. Laheld, "Equivariant adaptive source separation," *IEEE Trans. on Signal Processing*, vol. 44, pp. 3017 - 3030, 1996.

[2] P. Comon, "Independent component analysis - a new concept?," *Signal Processing*, vol. 36, pp. 287-314, 1994.

[3] C. Jutten and J. Herault, "Blind separation of sources, part I: an adaptive algorithm based on neuromimetic architecture," *Signal Processing*, vol. 24, no. 1, pp. 1-10, 1991.

[4] J. Karhunen, E. Oja, L. Wang, R. Vigario, and J. Joutsensalo, " A class of neural networks for Independent Component Analysis," *IEEE Trans. on Neural Networks 8*, pp. 486 - 504 (1997).

[5] Oja, E., "The nonlinear PCA learning rule in Independent Component Analysis," *Neurocomputing 17*, pp. 25 - 45 (1997).

[6] L. Wang, J. Karhunen, and E. Oja, “A bigradient optimization approach for robust PCA, MCA, and source separation,” *Proc. 1995 IEEE Int. Conf. on Neural Networks*, Perth, Australia, November 1995, pp. 1684 - 1689.

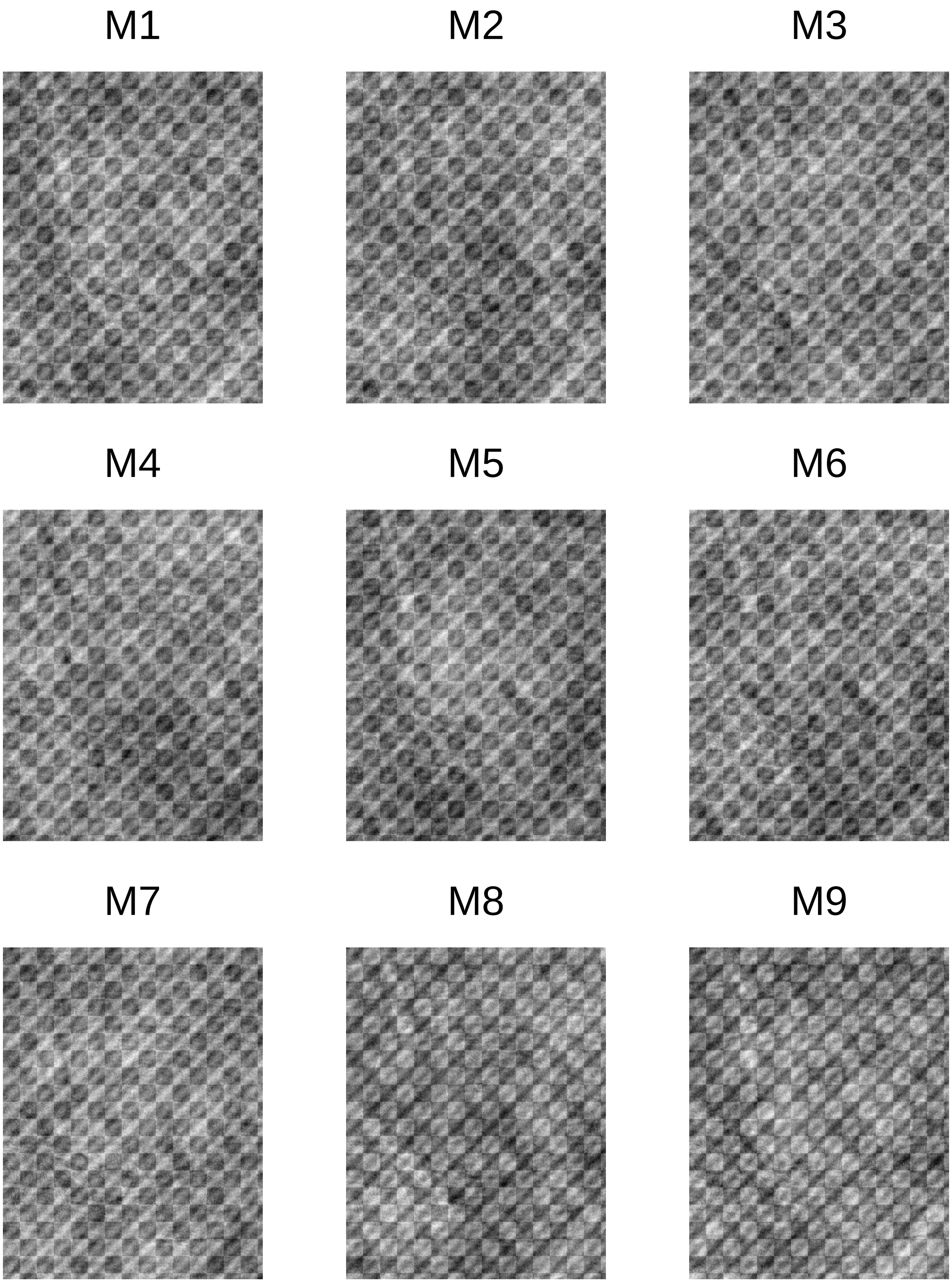


Figure 47: Mixtures of 9 natural images.

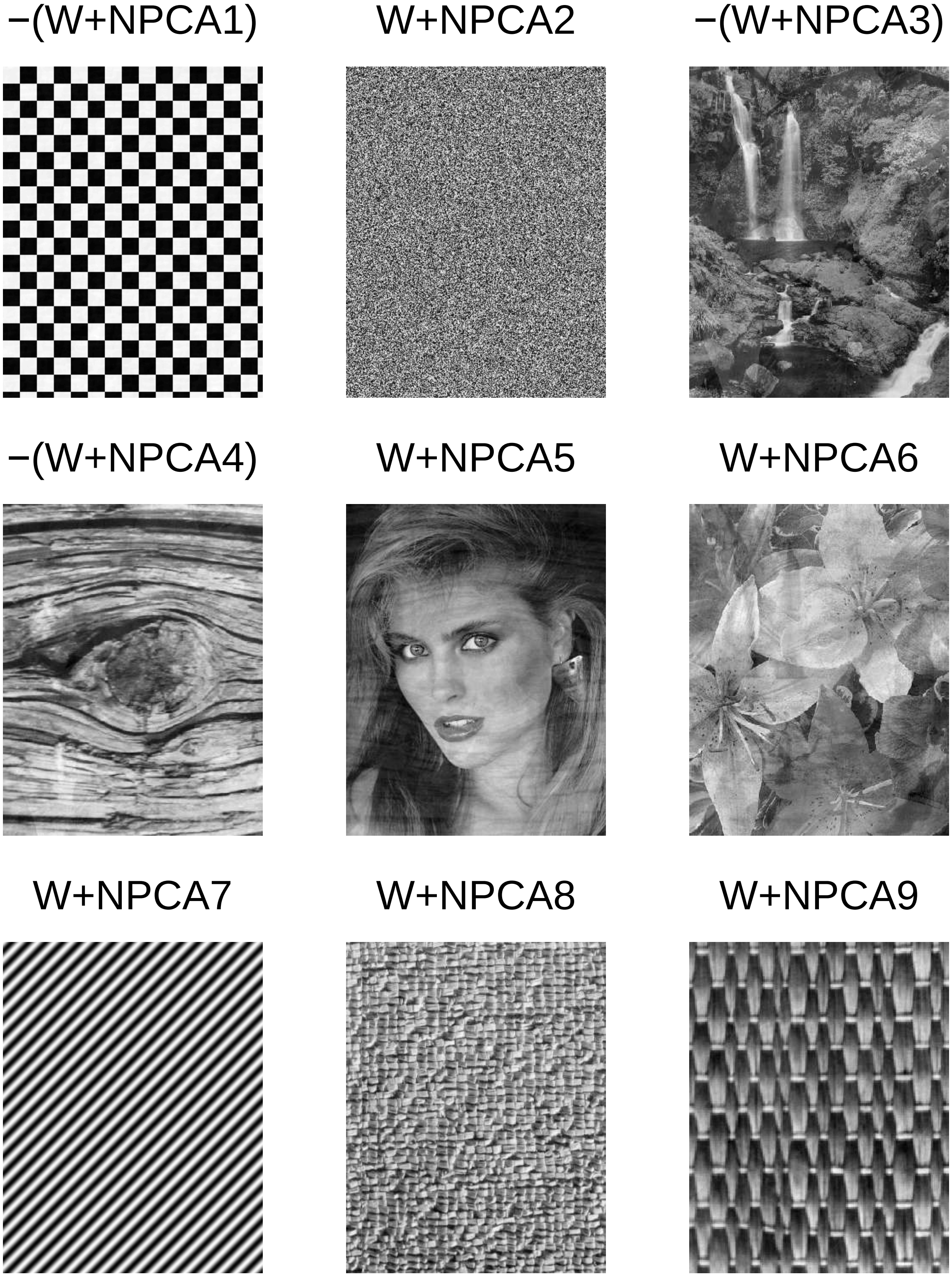


Figure 48: Separated images found by the 3-layer ICA network, using whitening and the nonlinear PCA algorithm.

32 One-unit and Fixed-point ICA Algorithms with Applications

Aapo Hyvärinen, Erkki Oja, Razvan Cristescu, Patrik Hoyer, Jarmo Hurri and Kimmo Kiviluoto

The starting point of this research project on Independent Component Analysis is the development of neural learning rules for a single unit [9,11,12]. Using these learning rules, a neural unit learns to separate from a multi-dimensional input signal one of the independent components, or a direction that has certain information-theoretic properties. This research can be considered a direct logical continuation of the original work by Oja on one-unit PCA [15], and shows clearly the connection between ICA and PCA. These one-unit learning rules are especially useful in exploratory data analysis, where they can be used to find single directions in the data space that show the most interesting and independent components in the data space.
For example, assuming that the observed signal $\mathbf{x}(t)$ is whitened, or sphered, we have obtained the following very simple learning rule for separating a sub-Gaussian independent component (i.e., an independent component of negative kurtosis):

$$\Delta \mathbf{w}(t) \propto \mathbf{x}(t) g(\mathbf{w}(t)^T \mathbf{x}(t)) - \mathbf{w}(t) \tag{89}$$

where the function g is a simple polynomial: $g(u) = au - bu^3$ with $a > 1$ and $b > 0$, $\mathbf{w}$ is the weight vector a neuron, and $\mathbf{x}$ is its input. This a very simple example of learning rules that are called Hebbian (or Hebbian-like), and which constitute one of the main paradigms in neural computing. In addition to learning rule (89), which separates sub-Gaussian independent components, one needs also a learning rule for separating super-Gaussian independent components (i.e., independent components of positive kurtosis). To achieve this, we have derived another learning rule:

$$\Delta \mathbf{w}(t) \propto b\mathbf{x}(t)(\mathbf{w}(t)^T \mathbf{x}(t))^3 - a\|\mathbf{w}(t)\|^4 \mathbf{w}(t). \tag{90}$$

where $a > 0$ and $b > 0$ are constants. This is also a Hebbian learning rule.
We have also developed a fast numerical method to implement the one-unit learning rules, the FastICA algorithm [10]. This method is based on a fixed-point iteration that usually speeds up the computations needed in ICA by a factor of 10 to 100. The FastICA algorithm is in a way a combination of the two preceding learning rules; the weight vector $\mathbf{w}$ is updated as follows:

$$\mathbf{w}^*(t) = E\{\mathbf{x}(\mathbf{w}(t-1)^T \mathbf{x})^3\} - 3\mathbf{w}(t-1) \tag{91}$$

$$\mathbf{w}(t) = \mathbf{w}^*(t)/\|\mathbf{w}^*(t)\| \tag{92}$$

where the expectation is, in practice, estimated using a large sample of $\mathbf{x}$ vectors. The difference from the preceding learning rules is basically that instead of using the inputs one-by-one, the FastICA algorithm first collects a batch of input data, and then uses all those data in the same learning step, in the computation of the average. The fast convergence of this fixed-point algorithm is illustrated in Figure 1, in which four images were recovered from four mixtures using altogether only 30 iterations. Another convenient property of this algorithm is that the same algorithm separates both super-Gaussian and sub-Gaussian independent components.

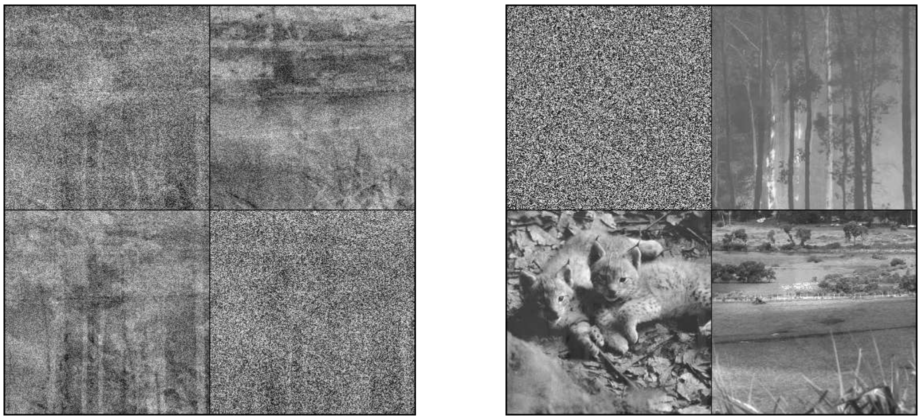


Figure 49: Three photographs of natural scenes and a noise image were linearly mixed to illustrate our algorithms. The mixtures are depicted on the left. On the right, the images recovered by the FastICA algorithm are shown. Only 7 iterations of the FastICA algorithm were required, on the average, for separating each image.

To separate several independent components, one can construct a network of several neurons, each of which learns according to the learning rules given above, and then add a feedback term to each of those learning rules.
The above learning rules, which are based on finding the extrema of kurtosis, have also been generalized for a large class of criteria of non-Gaussianity [12,2,5,9]. This means that the cubic non-linearity in the learning rules can be replaced by almost any other non-linearity. (Of course, some other changes are then also necessary.) Thus one obtains algorithms that often perform the ICA decomposition in a much more reliable and accurate way, according to such statistical criteria as asymptotic variance and robustness [3,9]. Also, the FastICA algorithm can be modified so that it works even when the data is corrupted by Gaussian noise [7].
The development of the FastICA algorithm has enabled us to apply ICA on data sets that are of a very high dimension. One example is image processing [1,8], where the FastICA algorithm has been used with success. Taking small windows of ordinary, real-life photographs, we decomposed the images into small components whose occurence is as independent from each other as possible. Some image components are depicted in Figure 2. Such a decomposition is likely to have interesting applications in image data compression, pattern recognition, and other domains of image processing. There are two reasons for this. First, such a decomposition resembles closely a so-called sparse coding. In sparse coding, one finds a coding method for the data that has certain interesting statistical properties, and fits with some neurophysiological measurements on the neural processing of sensory data. Second, the components found are reminiscent of so-called wavelets, which are used in some highly efficient techniques for image compression.
Based on the features given by ICA, we have developed a new method for image denoising [13,4] . This is based on modeling the noisy data by a noisy version of the ICA data model, and then estimating the original image by maximum likelihood estimation of the model. This results in the application of a (soft) thresholding operator on the features described above. Figure 3 shows an example of the application of this method, called sparse code shrinkage. The advantage of this method

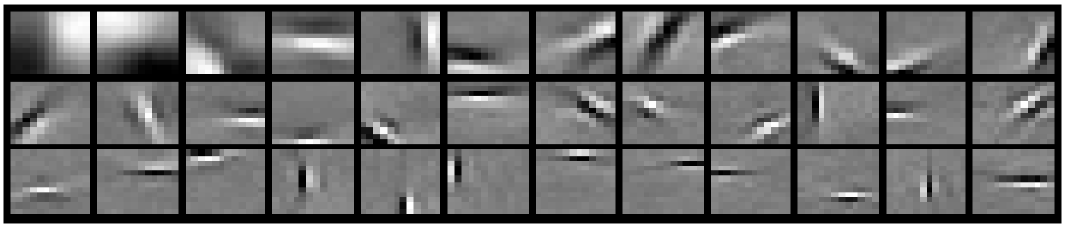


Figure 50: Image feature extraction by ICA. The FastICA algorithm was applied on image data, this time using small sub-windows of images as the data. Thus images were decomposed into hypothetical components whose occurrences are rather independent from each other. This figure shows some such components. One can see that the components define certain features that are often quite local, and resemble bar or edge detectors.

Figure 51: Image denoising by sparse code shrinkage. Using the theory of noisy ICA, one we have developed an image denoising method. This leads to a thresholding of the coefficients of the wavelet-like features shown in Figure 2.

over wavelet methods is that it is completely adaptive: both the features and the involved thresholding (shrinkage) functions are adapted to the statistical properties of the data.

We have also applied these methods on analysis of financial time series [14]. We used data that represented the simultaneous cash flow at several stores belonging to the same retail chain. ICA detected factors that affect the cash flow of all the stores. When the effect of these "fundamental factors" is removed, the impact of the actions of the management became more visible.

References

[1] J. Hurri, A. Hyvärinen, J. Karhunen, and E. Oja. Image feature extraction using independent component analysis. In *Proc. NORSIG'96*, pages 475–478, Espoo, Finland, 1996.

[2] A. Hyvärinen. A family of fixed-point algorithms for independent component analysis. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*

(ICASSP'97), pages 3917–3920, Munich, Germany, 1997.

[3] A. Hyvärinen. One-unit contrast functions for independent component analysis: A statistical analysis. In *Neural Networks for Signal Processing VII (Proc. IEEE Workshop on Neural Networks for Signal Processing)*, pages 388–397, Amelia Island, Florida, 1997.

[4] A. Hyvärinen. Independent component analysis in the presence of gaussian noise by maximizing joint likelihood. *Neurocomputing*, 22:49–67, 1998.

[5] A. Hyvärinen. New approximations of differential entropy for independent component analysis and projection pursuit. In *Advances in Neural Information Processing Systems 10*, pages 273–279. MIT Press, 1998.

[6] A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Trans. on Neural Networks*, 1999. To appear.

[7] A. Hyvärinen. Gaussian moments for noisy independent component analysis. *IEEE Signal Processing Letters*, 1999. To appear.

[8] A. Hyvärinen, R. Cristescu, and E. Oja. A fast algorithm for estimating overcomplete ICA bases for image windows. In *Proc. Int. Joint Conf. on Neural Networks*, Washington, D.C., 1999. submitted.

[9] A. Hyvärinen and E. Oja. Simple neuron models for independent component analysis. *Int. Journal of Neural Systems*, 7(6):671–687, 1996.

[10] A. Hyvärinen and E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7):1483–1492, 1997.

[11] A. Hyvärinen and E. Oja. One-unit learning rules for independent component analysis. In *Advances in Neural Information Processing Systems 9*, pages 480–486. MIT Press, 1997.

[12] A. Hyvärinen and E. Oja. Independent component analysis by general nonlinear Hebbian-like learning rules. *Signal Processing*, 64(3):301–313, 1998.

[13] A. Hyvärinen, E. Oja, P. Hoyer, and J. Hurri. Image feature extraction by sparse coding and independent component analysis. In *Proc. Int. Conf. on Pattern Recognition (ICPR'98)*, pages 1268–1273, Brisbane, Australia, 1998.

[14] K. Kiviluoto and E. Oja. Independent component analysis for parallel financial time series. In *Proc. ICONIP'98*, volume 2, pages 895–898, Tokyo, Japan, 1998.

[15] E. Oja. A simplified neuron model as a principal component analyzer. *J. of Mathematical Biology*, 15:267–273, 1982.

33 Robust Fitting by Nonlinear Neural Units

Erkki Oja and Liuyue Wang

A central problem in statistics is fitting a model which is linear in the parameters to a set of observation points. Examples are regression, curve fitting, time series modelling, digital filtering, system theory, and automatic control. The usual approaches are least squares (LS) or total least squares (TLS) regression. The difference between these approaches is shown in Fig. 52 in a simple line fitting example.
The TLS criterion is mathematically equivalent to finding the minor component of the input points, based on the eigenvector of the input covariance matrix corresponding to the smallest eigenvalue. In impulsive and colored noise environments, or in the presence of outliers, these methods are not optimal, however. Then robust fitting, based on a non-quadratic criterion, may give better results than the usual TLS.
The main objection to the use of robust fitting in practice has been a computational one: while the TLS criterion can be solved in closed form and the minor eigenvector can be computed with standard numerical techniques like the singular value decomposition (SVD), this is no longer true for more complicated criterion functions. An iterative gradient descent algorithm is necessary. Neural networks can be an advantage here [1,2,3].

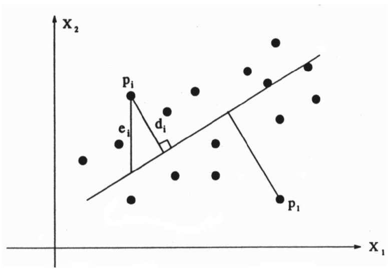


Figure 52: To fit a set of data points by a line, LS minimizes the sum of the squared lengths of the vertical distances e_i, whereas TLS minimizes the sum of the squared lengths of the distances d_i perpendicular to the estimated line.

Referring to Fig. 52, it holds $d_i = |\mathbf{w}^T\mathbf{x}_i|/\|\mathbf{w}\|$. Instead of using the TLS criterion, one may use an alternative criterion by minimizing the sum of certain functions of variable d_i instead of squares:

$$J_f(\mathbf{w}) = 1/N \sum_{i=1}^{N} f(d_i). \tag{93}$$

Generally, function $f(d_i)$ would be a monotonically increasing function of its nonnegative argument d_i. A meaningful choice is an even function, increasing slower than d_i^2. This will decrease the effect of strong outliers on the solution.
Replacing the finite sum in eq. (93) by the theoretical expectation of $f(\mathbf{w}^T\mathbf{x})$ and using a Lagrange multiplier for the constraint $\mathbf{w}^T\mathbf{w} = 1$ gives the following cost function:

$$J_L(\mathbf{w}, \lambda) = E\{f(\mathbf{w}^T\mathbf{x})\} + 1/2\lambda(1 - \mathbf{w}^T\mathbf{w}) \tag{94}$$

whose solution by an on-line gradient descent algorithm gives the following neural learning rule:

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \alpha_k[g(y_k)\mathbf{x}_k - (g(y_k)y_k + 1 - \mathbf{w}_k^T\mathbf{w}_k)\mathbf{w}_k] \tag{95}$$

where $y_k = \mathbf{w}_k^T\mathbf{x}_k$, $g(y)$ is the derivative of $f(y)$, and α_k is a positive learning rate. An especially suitable function for robust TLS fitting is $f(y) = \frac{1}{\beta}lncosh(\beta y)$, giving the usual sigmoid $g(y_k) = tanh(\beta y_k)$ as the neural network learning function. We call this the *Nonlinear Minor Component Analysis (NMCA)* algorithm.

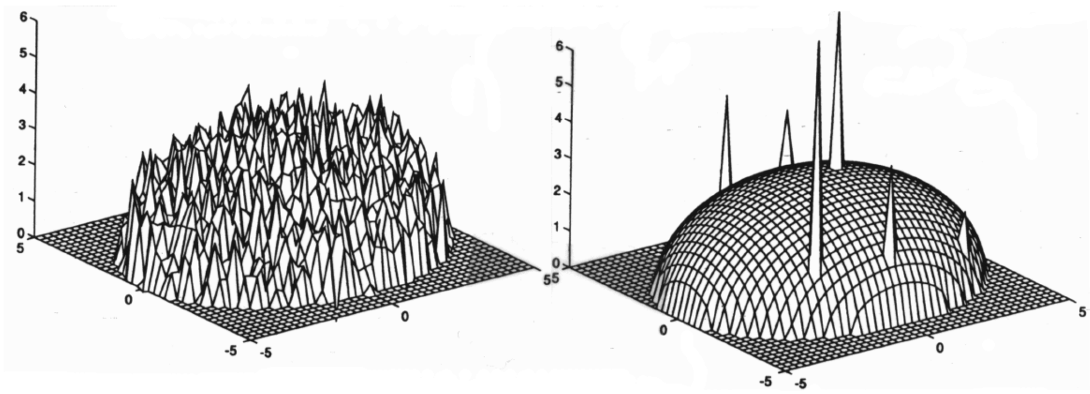


Figure 53: An experiment of surface fitting. Left: Gaussian noise. Right: outliers.

In an experiment of surface fitting [2], a data set D_x was used

$$D_x = \{(x_i, y_i, z_i), i = 1, ..., 993)\}$$

coming from an ellipsoid

$$0.04x^2 + 0.0625y^2 + 0.1111z^2 = 1.$$

Gaussian noise or six strong outliers were added to the sample points, as shown in Fig. 53. Like in line fitting, the problem is now to fit a parameterized model $w_1x^2 + w_2y^2 + w_3z^2 = 1$ to the point set D_x by estimating the parameters w_1, w_2, w_3. The results indicate that the error in the estimated parameters using the Nonlinear MCA algorithm (95) was about one third of the error obtained with conventional LS estimation in the Gaussian noise case and about 6 per cent in the case of outliers.

References

[1] L. Xu, E. Oja, and C. Y. Suen. Modified Hebbian Learning for Curve and Surface Fitting. *Neural Networks 5*, pp. 441–457 (1992).

[2] Oja, E. and Wang, L.: Robust fitting by nonlinear neural units. *Neural Networks 9*, pp. 435 - 444 (1996).

[3] Oja, E. and Wang, L.: Neural fitting: robustness by anti-Hebbian learning. *Neurocomputing 12*, pp. 155 - 170 (1996).

34 Analysis of Independent Components in EEG and MEG

Ricardo Vigário, Jaakko Särelä, and Erkki Oja

Without any doubt, the brain is among the most intriguing and complex systems ever studied by human-kind. In an attempt to give a plausible explanation to the *why's* and *how's* of human perception and cognition, many conjectures have been formulated and theories have been tested throughout centuries. In a bootstrapping (reinforced) manner, the discoveries made on the human brain are leading into the formulation of more efficient computational methods which in turn make it possible to design new signal processing tools for better extraction of information from brain data. Some of the most promising such tools are in the field of artificial neural networks, of which the independent component analysis (ICA) algorithm of this project is a good example.

The challenges presented to the signal processing community by the completely non-invasive electro- and magnetoencephalographic recordings from the human brain may be divided in two classes, one dealing with the identification and removal of artifacts from the recordings, and another the understanding of the brain signals themselves (see the Table below). The amplitude of the artifactual disturbances may well exceed that of brain signals, turning the analysis of brain activity into a very hard process. Moreover, artifacts may present strong resemblance to some physiological brain responses, bringing an erroneous interpretation of the recording.

Artifacts	*Brain signals*
Ocular artifacts	Evoked responses (e.g. auditory, somatosensory, visual, ...)
Myographic activity	Spontaneous rhythmic activity
Externally induced artifacts	Abnormal brain behavior (e.g. epileptic seizures, infarction, ...)

Over the past 3 years, combining the expert efforts from the Laboratory of Computer and Information Science, and the Brain Research Unit (both from the Helsinki University of Technology), we have shown that ICA techniques are very effective in helping to solve the problem of the extraction of artifacts from electroencephalographic and magnetoencephalographic recordings (EEG and MEG, respectively) [2,3] , enabling a better appreciation of these recordings by the physician.

Figure 54 presents a sample of the 122–channel MEG recordings, showing brain activity corrupted by a considerable amount of artifacts produced by eye saccades and blinks, head muscle activity and the cardiac cycle. The last three electrical signals were not used in the experimental setup, but are plotted for validation purposes. The artifacts, extracted using the FastICA algorithm (see section on ICA fixed-point algorithms), are shown in Fig. 55. Note that not only the strong corruptive signals (i.e. the muscle and eye activity) are correctly extracted, but even very weak artifactual signals are clearly isolated (IC4 and IC6 correspond to the cardiac cycle, and a digital watch, present in the shielded measuring room).

It is common to use event related activity as an entry level to the study of the human brain's functioning. This activity is time-locked to a particular stimulus, that may

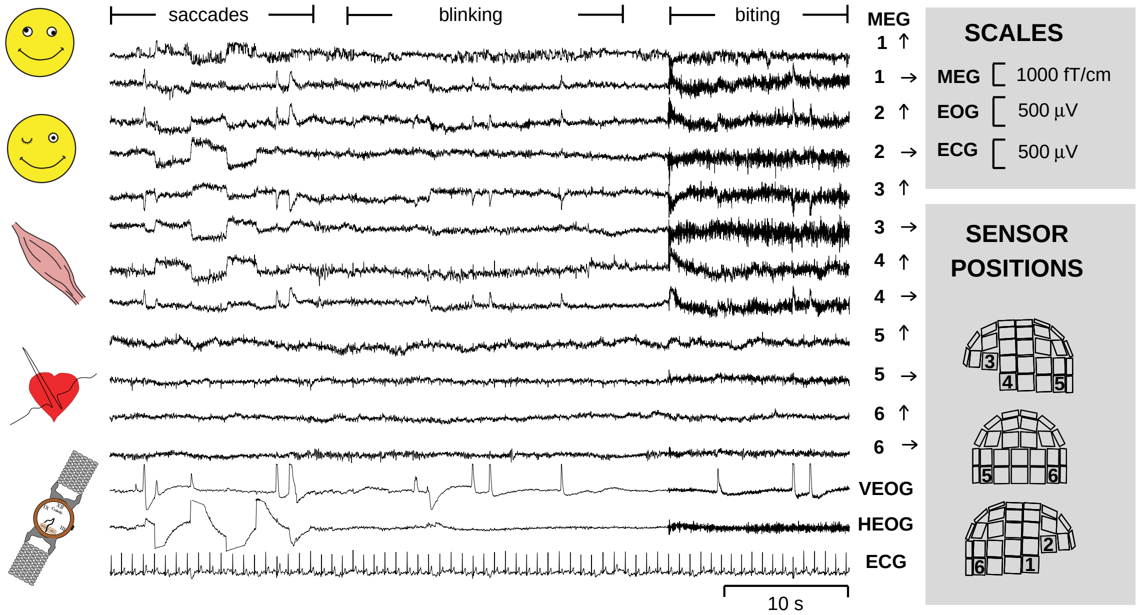


Figure 54: A sample of the 122-channel MEG recordings. For each of the 6 positions shown, the two orthogonal directions of the sensors are plotted (see [3] for further details).

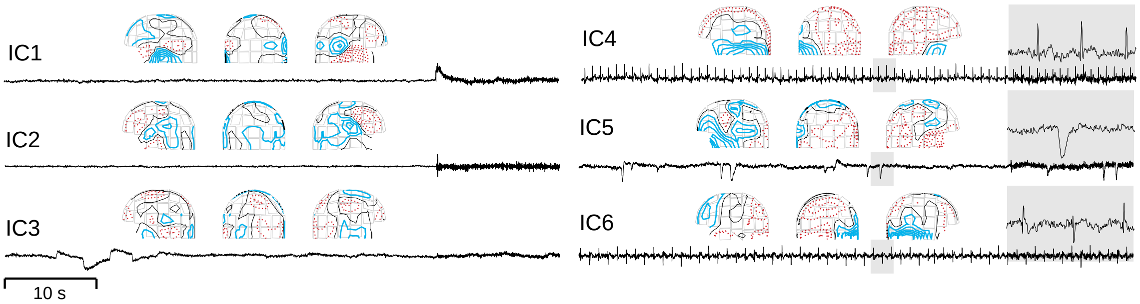
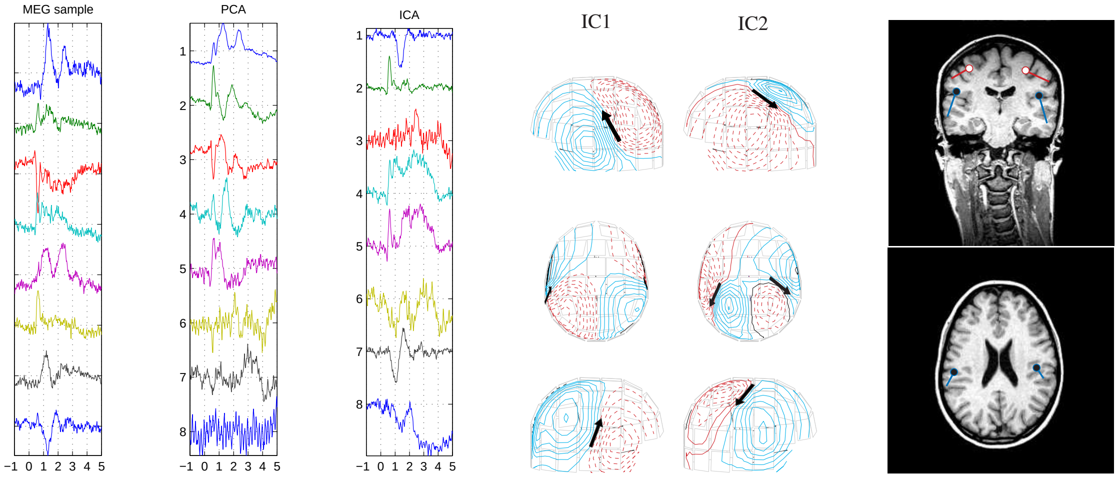


Figure 55: Six independent components extracted from the MEG data. For each component the left, back and right views of the field patterns are shown – full lines stand for magnetic flux coming from the head, and dotted lines the flux inwards. (from [3])

be of auditory, somatosensory, or visual type. Brain responses to the stimulation present minimal inter-individual differences to a particular set of stimulus parameters. In order to understand the physiological origins of the event related activity, it may be desirable to decompose the complex brain response into simpler elements, possibly easier to model and to localize their neural sources. In addition, the separation of multi-modal responses to complex stimuli, may represent a hard task to conventional methods, but is surely of capital importance, due to the diversity of stimulus modalities used in the perception of the real world.
Figure 56 shows a sample of the brain magnetic responses to a combined auditory and somatosensory stimulation *a*). The complex signals obtained are not resolved through PCA projection *b*), but rather well using an ICA approach *c*). The field patterns corresponding to the first two independent components (two columns of the estimated mixing matrix **A**), are depicted in frame *d*). The different colors used stand for the different orientations of the magnetic flux on the sensor plane. Using this information, together with some dipole source modeling, we reach the



(a) (b) (c) (d) (e)

Figure 56: Results of the application of FastICA to the average brain MEG responses to a vibrotactile stimulation. The last frame show the localization of the brain sources superimposed onto an MRI scan. (adapted from [4]).

localizations present in frame *e*), superimposed onto an MRI scan of the subject. The locations found have a perfect agreement to the ones suggested by conventional neurolophysiological theories. Further information on these experiments may be seen in [4,5] . Promising results were as well obtained in the analysis of non-averaged evoked responses [1] .
The results reported in this section showed that ICA is not only a very efficient artifact removal tool both for EEG and MEG, but as well gives very promising results when dealing with the more demanding problem of extracting information from the brain's own activity. The global list of publications, at the end of this report, contains further references to this work. The ones in this section should give a good starting point to the understanding of the results achieved within the project.

References

[1] J. Särelä, R. Vigário, V. Jousmäki, R. Hari, and E. Oja. In *4th Int. Conf. on Functional Mapping of the Human Brain (HBM'98)*, Montreal, Canada, 1998.

[2] R. Vigário. *Electroenceph. clin. Neurophysiol.*, 103:395–404, 1997.

[3] R. Vigário, V. Jousmäki, M. Hämäläinen, R. Hari, and E. Oja. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Neural Information Processing Systems 10 (Proc. NIPS'97)*, Cambridge MA, 1998. MIT Press.

[4] R. Vigário, J. Särelä, V. Jousmäki, and E. Oja. In *Proc. Int. Workshop on Independent Component Analysis and Blind Separation of Signals (ICA'99)*, Aussois, France, January 1999.

[5] R. Vigário, J. Särelä, and E. Oja. In *Proc. Int. Conf. on Artificial Neural Networks (ICANN'98)*, Skövde, Sweden, September 1998.

35 Extensions of the Basic Source Separation Problem

Petteri Pajunen, Juha Karhunen, Aapo Hyvärinen, Harri Lappalainen, and Simona Mălăroiu

35.1 Nonlinear mixing of the sources

In the basic Independent Component Analysis (ICA) model

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) \tag{96}$$

it is assumed that the M unknown source signals are linearly mixed into M different known mixtures. Here $\mathbf{s}(t)$ denotes the M-vector containing the M source signals at time t. The matrix $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_M]$ is a constant $M \times M$ mixing matrix whose elements are the unknown coefficients of the mixtures. The columns $\mathbf{a}_i$ of $\mathbf{A}$ are the basis vectors of ICA, and $\mathbf{x}(t)$ is the M-dimensional tth data vector made up of the mixtures at discrete time (or point) t.

In realistic applications the linearity assumption of the simple basic model (96) is not necessarily valid. Since ICA defines a linear transformation $\mathbf{B}$ which makes the components of the random vector $\mathbf{y}(t) = \mathbf{B}\mathbf{x}(t)$ as independent as possible, it is natural to consider more general transformations which have the same effect.

The Self-Organizing Map (SOM) can be used to define a nonlinear transformation which approximately estimates the probability density of the input data. The weight vectors of SOM are distributed proportionally to the input vector density. Using certain learning rules this relationship is accurate, and the distribution of the SOM weight vectors is asymptotically the same as the distribution of the input vectors. This forces each weight vector to have the same probability of 'winning' and therefore the distribution on the converged map is uniform. By using a rectangular map, the output vector coordinates become approximately statistically independent [1]. We have successfully applied this property of the SOM to the blind source separation problem when the source signals have a flat (sub-Gaussian) distribution, and the nonlinear mixing function is not too nonlinear [2]. The restriction of using SOM is that the source densities are implicitly modeled as uniform densities. If there is prior knowledge of the source densities, this can be used to improve the results. The generative topographic map (GTM) allows to do this. We applied this to the nonlinear ICA problem with improved results compared to SOM [7].

Even though this method has some limitations, its advantage is that it is truly neural, contrary to the few other existing approaches to the generally very difficult nonlinear blind separation problem.

We have also studied the theoretical questions that arise when considering nonlinear ICA. Especially we have shown that the solution to nonlinear ICA always exists but is highly non-unique. We have also developed a set of conditions which lead to a unique solution [9].

35.2 Binary sources

Another restriction of the basic ICA model is the assumption that the number of available mixtures equals to the number of source signals. If the source signals are continuous, it is generally impossible to separate more sources than mixtures, because the solution of the blind separation problem becomes highly nonunique. However, assuming that all the source vectors are binary and all the mixtures are different, the mixing transformation is one-to-one, and in theory it then becomes possible to separate the sources. In the special case of two mixtures the separation can be achieved by computing the convex hull of the observed mixtures [3]. It can be shown that the convex hull uniquely determines the basis vectors of ICA (and the number of them) under mild assumptions. The sources can then be easily separated when the basis vectors are known.

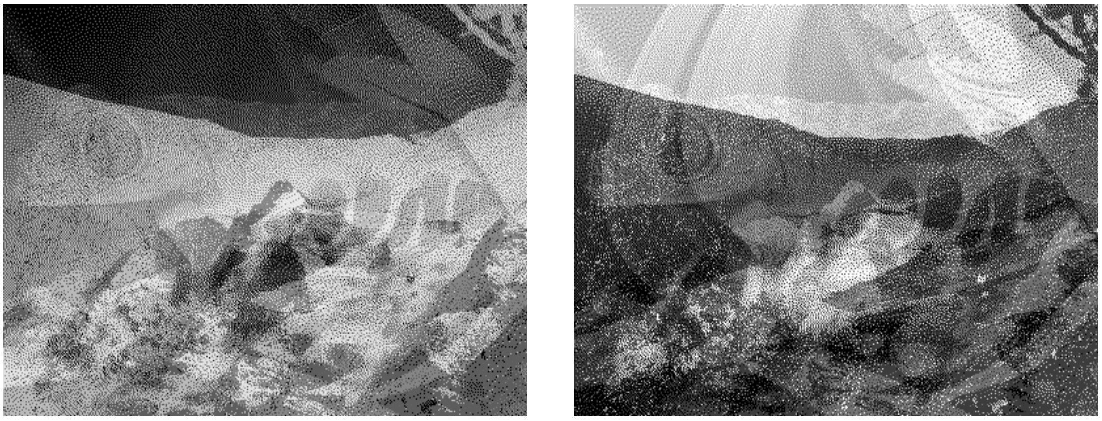


Figure 57: Noisy mixture images.

An example of blind separation of binary sources is given in figures 57 and 58 highlighting the possibility of separating binary signals from less mixtures than sources. In figure 57, two noisy linear mixtures of four binary images are shown. A binary source separation algorithm developed by us was applied to these mixtures producing the separated images shown in figure 58.

35.3 The effect of noise, correlation, and various network structures to separation results

In a joint project with the laboratory of Artificial Brain Systems, RIKEN research institute, Japan, we have considered some other in practice interesting extensions of the basic blind source separation problem. Such extensions have been outlined and discussed in an invited tutorial review paper [4]. We have in particular considered what happens in different neural network structures when the number of source signals is different from the number of sources and/or outputs of the network, and proposed various methods for handling such situations. We have also studied the ability of the networks to separate correlated sources, and the effect and removal or suppression of noise in context with blind separation. Two journal papers [5,6] summarize the results achieved in this joint project on these topics.

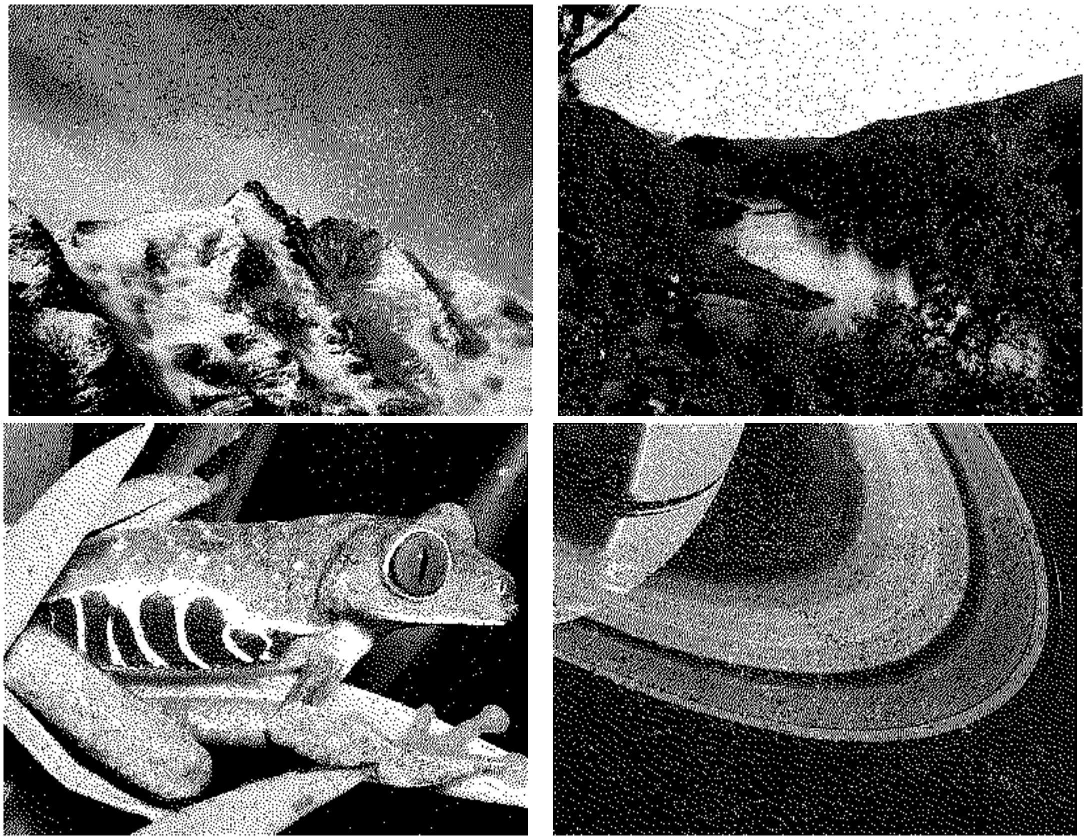


Figure 58: Separated images.

Later on, we have shown [16] that if there is additive noise present in the basic ICA/BSS model (96), the optimal solution of the problem in fact depends nonlinearly on the observed mixture vectors $\mathbf{x}(t)$. Computationally efficient approximations to the optimal maximum likelihood solution of this problem have been derived in various situations in [16].

35.4 Local ICA methods

In standard ICA, a linear data model (96) is used for a global description of the data. Even though linear ICA yields meaningful results in many cases, it can provide a crude approximation only for nonlinear data distributions. In [10], a new structure is proposed, where local ICA models are used in connection with a suitable clustering algorithm grouping the data. The clustering part is responsible for an overall coarse nonlinear representation of the underlying data, while linear ICA models of each cluster are used for describing local features of the data. The goal is to represent the data better than in linear ICA while avoiding computational difficulties associated with nonlinear ICA. In first experiments with such a local ICA method, we have used simple K-means clustering. The proposed method performs well for natural image data, yielding meaningful local features with suitable preprocessing [10].

35.5 Generalization of ICA using complexity and coding

It is possible to generalize independent component analysis by considering representations of the observed mixtures which can be coded using as few bits as possible. Equivalently, we can look for representations that have a minimum complexity. Choosing a linear representation and measuring the complexity using entropy, we obtain the same approach as in ICA where mutual information is minimized [8]. The generalization here is conceptual and is of fundamental importance. It allows principled application of ICA to *any* data instead of noiseless data containing linear mixtures of strictly independent sources. The general form of the complexity measure serves as a framework for true extensions. The entropy can be replaced by any other coding measure, which can be chosen quite freely. Using compression algorithms to approximately measure the codelength yields improved results compared to standard ICA algorithms [8]. Using principal component analysis to measure the complexity leads to new algorithms as well [11, 12].

35.6 Bayesian learning

In modeling there is a trade-off between the flexibility of models and robustness against overfitting. Too simple a model is not able to capture all the regularities and structure of the data, but too complex a model overfits, i.e., learns also the coincidental noise always present in real data.

Bayesian approach to learning solves the trade-off by finding the most probable model. It is closely related to information theoretically motivated approaches which minimize the description length of the data, because the description length is defined to be the minus logarithm of the probability. Minimal description length thus means maximal probability.

In practice, Bayesian learning involves approximating the posterior density of the models. This has been done using a recently developed method called ensemble learning, where a simple parametric approximation is fitted to the posterior density by minimizing the Kullback-Leibler distance. The method has been applied to linear ICA in [13]. A nonlinear extension, where the nonlinear mapping from the sources to the observations is modeled by a multi-layer perceptron (MLP) network, has been studied in [14]. The methods for using ensemble learning with MLP networks have been developed in [15] using an information theoretically motivated approach.

References

[1] P. Pajunen. Nonlinear independent component analysis by self-organizing maps. In C. von der Malsburg et al., editors, *Artificial Neural Networks – ICANN'96*, pages 815–819. Springer, 1996.

[2] P. Pajunen, A. Hyvärinen, and J. Karhunen. Nonlinear blind source separation by self-organizing maps. In S. Amari et al., editors, *Progress in Neural Information Processing (ICONIP-96)*, pages 1207–1210. Springer, 1996.

[3] P. Pajunen. An algorithm for binary blind source separation. Technical Report A36, Lab. of Computer and Information Science, Helsinki University of Technology, 1996.

[4] J. Karhunen. Neural approaches to independent component analysis and source separation. In *Proc. of the 4th European Symposium on Artificial Neural Networks (ESANN'96)*, pages 249–266, Bruges, Belgium, April 1996.

[5] J. Karhunen, A. Cichocki, W. Kasprzak, and P. Pajunen. On neural blind separation with noise suppression and redundancy reduction. *Int. J. of Neural Systems*, vol. 8, no. 2, pp. 219-237, 1997.

[6] A. Cichocki, J. Karhunen, W. Kasprzak, and R. Vigario. On neural blind separation with unequal numbers of sensors, sources, and outputs. *Neurocomputing*, vol. 24, pp. 55-93, February 1999.

[7] P. Pajunen and J. Karhunen. A maximum likelihood approach to nonlinear blind source separation. In *Proc. Int. Conf. on Artificial Neural Networks (ICANN'97)*, pages 541–546, Lausanne, Switzerland, October 1997.

[8] P. Pajunen. Blind source separation using algorithmic information theory. *Neurocomputing*, vol. 22, pp. 35-48, 1998.

[9] A. Hyvärinen and P. Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks*, vol. 12, no. 3, pp. 429-439, 1999.

[10] J. Karhunen and S. Mălăroiu. Local independent component analysis using clustering. In *Proc. Int. Workshop on Independent Component Analysis and Signal Separation (ICA'99)*, pp. 43–48, Aussois, France, January 1999.

[11] P. Pajunen. Blind source separation of natural signals based on approximate complexity minimization. In *Proc. Int. Workshop on Independent Component Analysis and Signal Separation (ICA'99)*, pp. 267-270, Aussois, France, January 1999.

[12] A. Ypma and P. Pajunen. Rotating machine vibration analysis using second-order independent component analysis. In *Proc. Int. Workshop on Independent Component Analysis and Signal Separation (ICA'99)*, pp. 37-42, Aussois, France, January 1999.

[13] H. Lappalainen. Ensemble learning for independent component analysis. In *Proceedings of the ICA'99*, pages 7–12, Aussois, France, January 1999.

[14] H. Lappalainen and X. Giannakopoulos. Multi-layer perceptrons as nonlinear generative models for unsupervised learning: a Bayesian treatment. Submitted for ICANN'99.

[15] H. Lappalainen. Using an MDL-based cost function with neural networks. In *Proceedings of the IJCNN'98*, pages 2384–2389, Anchorage, Alaska, May 1998.

[16] A. Hyvärinen. Independent component analysis in the presence of Gaussian noise by maximizing joint likelihood. *Neurocomputing*, vol. 22, pp. 49–67, 1998.

36 Intelligent Process Data Analysis

**Olli Simula, Jussi Ahola, Esa Alhoniemi,
Johan Himberg, Pekka Hippeläinen, Jaakko Hollmén,
Juha Parhankagas, Jukka Parviainen and Juha Vesanto**

Analysis and control of complex nonlinear processes constitutes a difficult problem area in many practical applications. In complicated systems it is not possible to model the system analytically. In this case analysis of the available process data is the only possible approach. The data analysis begins with acquisition of all available data describing the system. The quality of the data is improved by removing noise and clear-cut errors. After this, based on process knowledge some variables may be combined to form new ones which are more useful from the problem point of view. The process, depicted in Figure 59, is repeated iteratively in order to find the most important variables relevant to the problem.

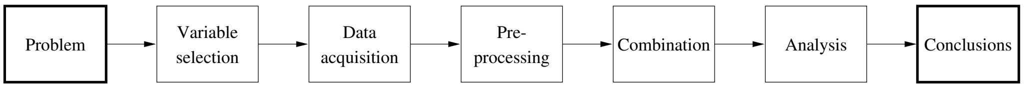


Figure 59: Data analysis process. In practice this sequential schema is iterated over and over again.

The SOM has proven to be a powerful tool to aid in the analysis due to its ability to form a visual illustration of data with respect to the selected set of variables. The SOM has the desirable feature of describing the nonlinear relationships between the large number of parameters and variables phenomenologically. Because the SOM algorithm performs a topology preserving mapping from the high-dimensional space to map units, it can also serve as a clustering tool of high-dimensional data. The SOM has also a capability to generalize, i.e. the network can interpolate between previously encountered inputs.
In this research project, several industrial processes have been analyzed in close cooperation with industrial partners. These include companies in steel and forest industry as well as design and consulting. Part of the work has been carried out in the TEKES Technology Program on Adaptive and Intelligent Systems Applications. In addition, work has been carried out in the Brite-Euram project, Application of Neural Networks Based Models for Optimization of the Rolling Processes (NEUROLL), which concentrates on improving steel manufacturing processes: using the methodology presented above, it is possible to investigate complex dependencies between incoming raw materials, process parameters at different stages, and quality parameters of the final product. Some case studies will be described in more detail below.

36.1 Pulp Mill

In a case study, behavior of a continuous pulp digester was analyzed. An illustration of the digester and separate impregnation vessel is shown in Figure 60. Wood chips and cooking liquor are fed into the impregnation vessel. After the impregnation, the chips are fed into the digester. At the top of the digester, they are heated to cooking temperature using steam, and the pulping reaction starts. During the cook, the chips slowly move downwards the digester. The cooking ends at extraction screens by displacement of hot cooking liquor by cooler wash liquor, which is injected to the digester through bottom nozzles and bottom scraper. The liquor moves counter-current to the chip flow and carries out washing of the chips.

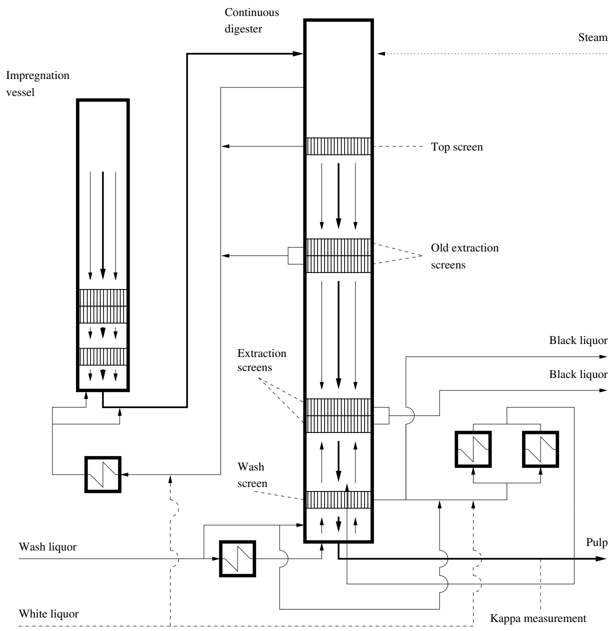


Figure 60: The continuous digester and the impregnation vessel. The cooking and wash liquor flows are marked by thin lines and the chip flow by thick line.

Digester operation problems indicated by drops of pulp consistency in the digester outlet were the starting point for the analysis. In those situations, end product quality variable (kappa number) values were smaller than the target value.

Measurement data were obtained from the automation system of the mill. The analysis was started with several dozens of variables which were gradually reduced down to six most important measurements during the data analysis process. The data used in the experiments consisted of three separate measurement periods during more than one month of normal pulping operation.

The periods were segmented by hand in such a way that they mainly consisted of faulty situations of the process. The production speed was required to be constant. During the measurement periods there were no significant errors in the measurements. Process delays between signals were compensated using beforehand known digester delays. In Figure 61 the six signals and production speed of the fiber line are shown. The three segmented parts are shown by solid line and the parts that were left out of the analysis by dotted line.

In Figure 62, the component planes of a 17 by 12 units SOM trained using signals of Figure 61 are presented. Five of them depict behavior of the digester and the last one is the output variable, the kappa number. The most problematic process

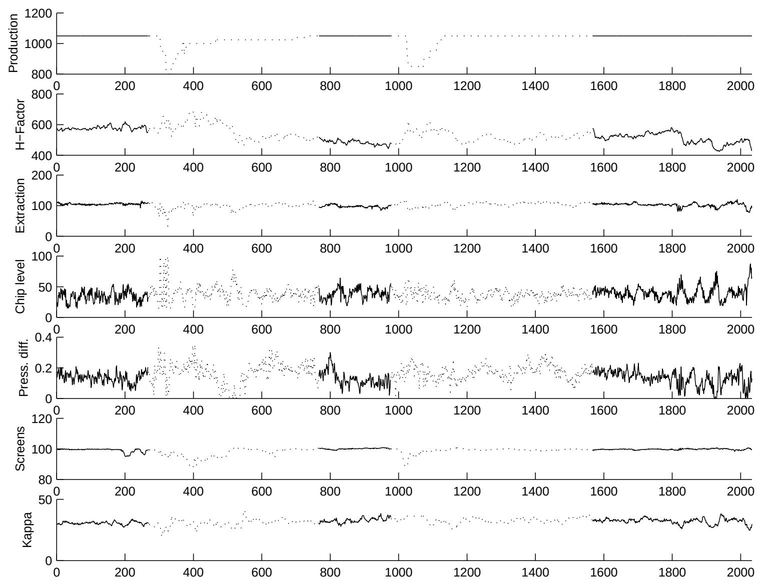


Figure 61: Measurement signals of the continuous digester. The analyzed parts are marked by solid line and the parts that were ignored by dotted line.

states are mapped to top left corner of the SOM: the model vectors in that part of the map have too low kappa number value.

Correlations between the kappa number and other variables were studied using Figure 63, where the SOM of Figure 62 has been presented using continuous color coding. The colors assigned to map units are shown in the top left corner of Figure 63. The five scatter plots are based on *model vector component values* of the SOM. They all have the values of kappa number on the x-axis and the other five variables on y-axis.

The scatter plots indicate that *in the faulty states* denoted by violet color there is weak correlation between kappa number and H-Factor, which is the variable used to control the kappa number. Otherwise there is a negative correlation as might be expected. On the other hand, the variables *Extraction* and *Chip level* seem to correlate with the kappa number. Also, the values of *Press. diff.* are low and the value of variable *Screens* (which during the analysis was noticed to indicate sensitivity of digester faults) is high.

The interpretation of the results is that in a faulty situation, the downward movement of the chip plug in the digester slows down. The plug is so tightly packed at the extraction screens that the wash liquor cannot pass it as it should. There are two consequences: the wash liquor slows down the downward movement of the plug and the pulping reaction does not stop. Because the cooking continues, the kappa number becomes too small. In addition, the H-factor based digester control fails: in the H-factor computation, cooking time is assumed to be constant, while in reality it becomes longer due to slowing down of the chip plug movement.

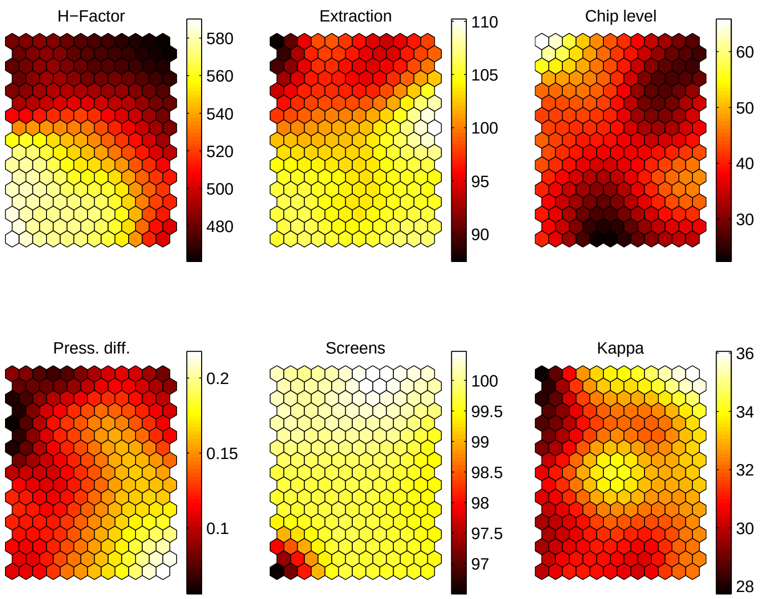


Figure 62: Component planes of the SOM trained using six measurement signals of the digester.

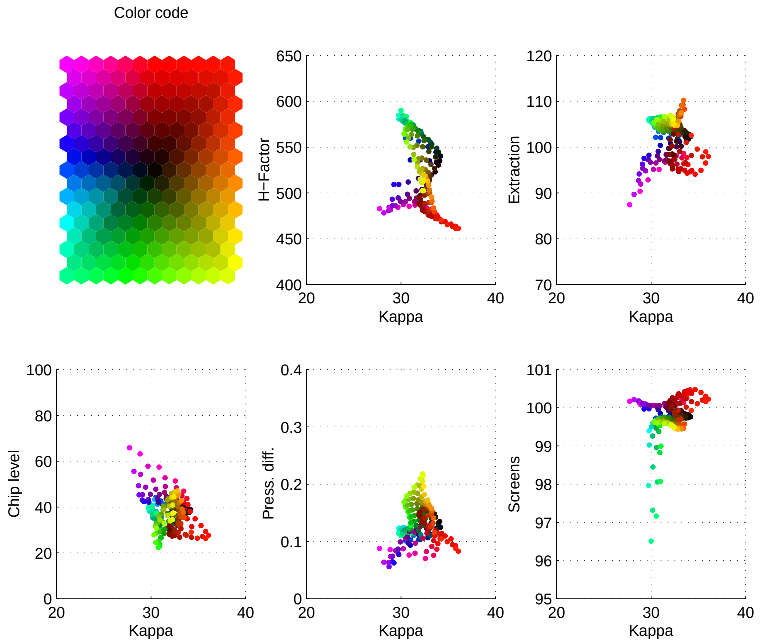
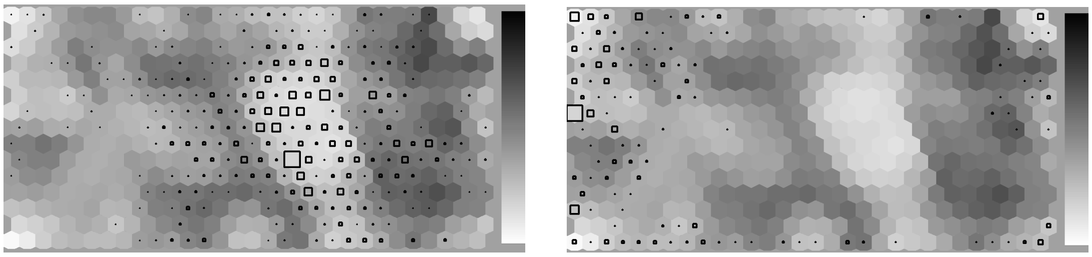


Figure 63: Color map and five scatter plots of model vectors of the SOM. The points have been dyed using the corresponding map unit colors.

36.2 Pulp and Paper Mills Technology

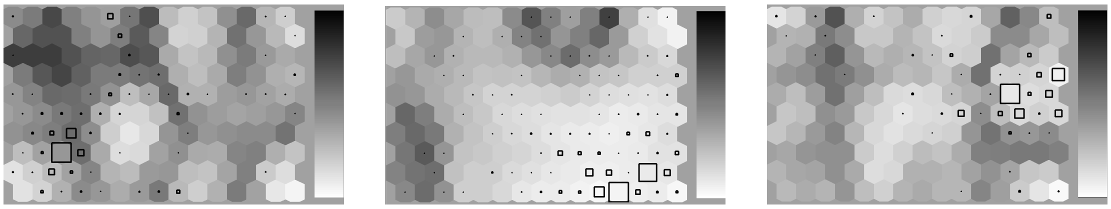
In this case study, the technology of pulp and paper mills all over the world was studied. The data was divided to three separate sets: information of the mill itself, its paper machines, and its pulp production. One SOM was trained for each of the three sets, and a fourth SOM was built using the combination of BMU coordinates of the input data on the three low level maps.
Figures 64a and 64b show the unified distance matrix (u-matrix) of the combined map, and the distribution of Chinese and Scandinavian paper mills on the map. The two mill sets are easily separable, although there was no geographic information present in the data. It can also be seen that in the area where the Chinese mills are, the values of the u-matrix are very low. That is, the variation between weight vectors in that area is low, which means that most of the Chinese mills resemble each other.
Figures 65a, 65b and 65c show the distribution of Chinese paper mills on the three low level maps. Also from these figures it is apparent that most of the Chinese mills are centered on a single area of the map. Taking a look at the weight vectors in these areas, it can be seen that the typical Chinese paper mill has small capacity, a large number (e.g. 4) of paper machines and it most probably produces printing/writing paper. The paper machines are small, slow and the paper weight is low. The pulp is produced chemically. Scandinavia, on the other hand, represents a technologically advanced region. The mills are new, they have big-capacity paper machines and the majority produces printing/writing papers or pulp.



(a) Chinese

(b) Scandinavian

Figure 64: Chinese (a) and Scandinavian (b) paper mills on the u-matrix of the combined map.



(a) Mills (b) Paper machines (c) Pulp production

Figure 65: Chinese paper mills on the three low level maps: mills (a), paper machines (b) and pulp production (c).

36.3 Other Applications

In a project completed in 1998, a process control software, SOM-Line, based on SOM was implemented. The software was a tool for monitoring functions of a galvanizing line of steel. The SOM is first trained by classified measurements. After training the software is capable of monitoring change in process status illustratedly in real time. If necessary the process can be adjusted based on information supplied by the software.

In a project with Suomen Perusmetalli continuous casting is being investigated. Continuous casting is a large scale industrial method for producing steel slabs for further refinement, e.g. for hot rolling. There may come up surface ruptures on the steel slab during the casting. The ruptures lower the product quality, and sometimes they may cause a breakthrough of liquid steel that leads to a long maintenance pause in the casting plant. The casting process is, however, monitored by thermocouples inside the mold. The obtained temperature data is analyzed using SOM-based visualization tools in order to get efficient features for rupture and breakthorugh warnings in the automation system.

The NEUROLL ("Application of Neural Network based Models for Optimisation of the Rolling Process") in an EU-financed project, the objective of which is to improve the efficiency of the production and the end quality of the hot rolled steel products. This is achieved by detecting complex relationships between measured or calculated process parameters and input/output variables, using statistical data analysis methods. Furthermore, the process is modeled at some level and methods are developed for the process state visualization and monitoring, as well as for diagnosis purposes.

In our laboratory the study has concentrated in finding correlations between quality parameters (e.g., width and thickness deviation and surface defects) and other process parameters using traditional correlation analysis and the SOM-based methods. This task is almost done and the results are utilized in improving the process in the general level and finding the variables for the process modeling. So, this task is also in progress and several, mostly SOM-based, approaches for process state visualization and monitoring have been considered.

In a project with Metsäteho Oy, the Finnish forest research organisation, data collected by forest harvesters is analysed. Data consists of measurements from individual trees and forest areas as a whole. The main goal is to apply unsupervised clustering techniques to group similar forest areas together. This clustering could prove to be useful when controlling the resource management for Finnish sawmills.

References

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

[2] T. Kohonen. *Self-Organizing Maps*. Springer, Berlin, Heidelberg, 1995.

[3] T. Kohonen, E. Oja, O. Simula, A. Visa, and J. Kangas. Engineering applications of the self-organizing map. *Proceedings of the IEEE*, 84(10):27 pages, October 1996.

[4] O. Simula and J. Kangas. *Neural Networks for Chemical Engineers*, volume 6 of *Computer-Aided Chemical Engineering*, chapter 14, Process monitoring and visualization using self-organizing maps. Elsevier, Amsterdam, 1995.

[5] O. Simula, P. Vasara, J. Vesanto and R-R. Helminen. *Industrial Applications of Neural Networks*, chapter 4, The Self-Organizing Map in Industry Analysis, eds. L.C. Jain and V.R. Vemuri. CRC Press, 1999.

[6] J. Vesanto. SOM-Based Data Visualization Methods. *Intelligent Data Analysis*, April 1999 (to appear).

37 Adaptive Receivers Based on Self-Organizing Maps

Kimmo Raivio, Olli Simula, and Teuvo Kohonen

In this research project, new receiver structures based on the Self-Organizing Map (SOM) algorithm [2] have been developed. The SOM is used both as an adaptive decision device and to follow up error signals.

The SOM is a competitive neural network algorithm that produces localized responses to input signals and represents the topology of the input signal space over the network. Due to the active topology preserving property of the learning scheme, the SOM is able to adapt to time-varying situations. In communication systems, the signals are corrupted with various distortions caused by the transmission channel, interfering signal and noise (Fig. 66). These distortions can be adaptively compensated using the capability of on-line adaptation of the SOM [3, 5].

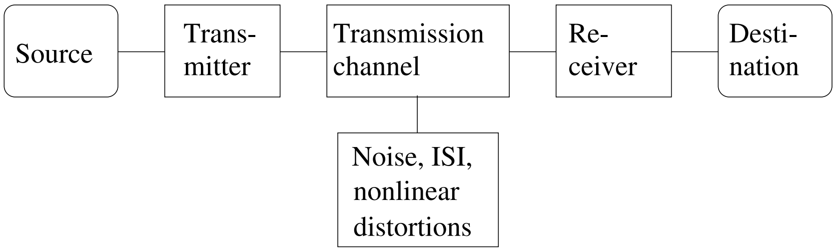


Figure 66: Communication system

When the SOM is used as an adaptive decision device, comparisons with conventional equalizers such as the linear equalizer and the decision feedback equalizer (DFE) [4] have been performed. The new structures were also compared with other neural methods like radial basis function (RBF) networks and multi-layer perceptrons (MLP) [1]. The performances of the neural equalizers and especially the SOM have been found to be better in nonlinear multipath channels and about equal in linear channels.

When the SOM has been used to follow up error signals, the actual idea has been to cancel interference. This task has been divided between following up the error distribution and finding out the error estimate. The error is approximately the same as the interference. Other sources of error are noise, intersymbol interference, wrong error estimates and detection errors due to the reasons mentioned before. The error distribution can be followed up, but the problem is how to predict the error. Some solutions have been found, but they do not provide satisfactory results. The performance has been compared with a pure detector without any kind of interference cancellation and with a receiver based on the RBF network.

In our research, the wanted signal has been of QAM (Quadrature-Amplitude Modulation) type as well as the interfering signal if it is present. The aim of the research has been to use the SOM as a building block of new adaptive receivers, which are able to compensate the nonlinear distortions or cancel the interfering signals.

37.1 Compensation of Nonlinear Distortions

Neural networks are an obvious choice for the compensation of nonlinearities, because the task is often such that either analytic solutions do not exist or they cannot be found. The networks can be trained to follow up distortions.
The Self-Organizing Map algorithm is used as an adaptive detector preceeded by the DFE (Fig. 67). In the conventional SOM the samples are classified one after each other, but the algorithm can also be made to accept input data as batches.

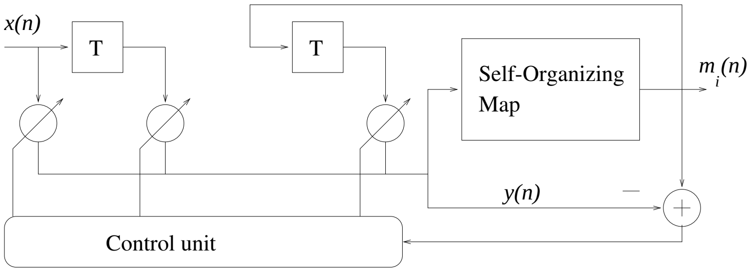
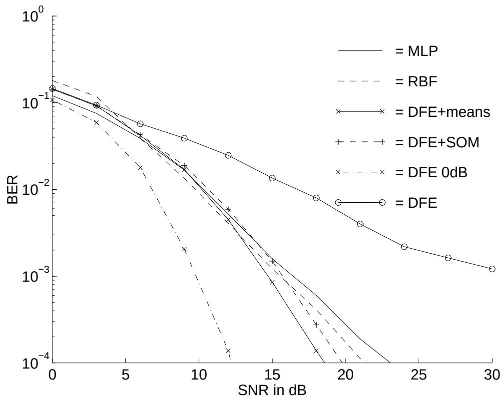
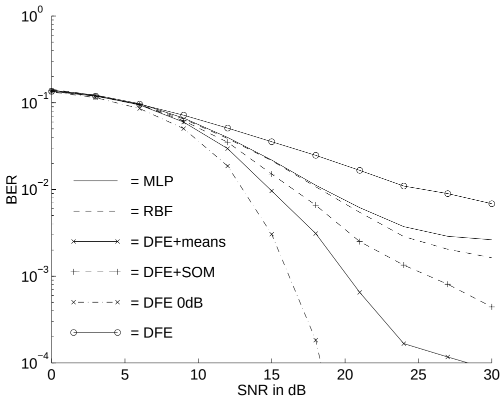


Figure 67: DFE and SOM in cascade

Usually, the nonlinear distortion is introduced by transmitter or receiver amplifiers, but other sources are also possible. More often the distortions are unwanted, but sometimes the signal is distorted on purpose. It is possible, that the distortion has been created in the receiver amplifier in order to cut off the amplitude peak values. When such a distortion is compensated, the signal to interference ratio (SIR) vs. bit-error-ratio (BER) of the equalizers in a two-path channel are as shown in Figure 68.



(a) 16-QAM signal and 6 dB decay



(b) 64-QAM signal and 3 dB decay

Figure 68: Compensation of nonlinear amplifier which decays the signal.

37.2 Interference Cancellation

The SOM has been used to interference cancellation by feeding detection errors into the map. The idea is to form an estimate of the error on the basis of previous error and signal values. A signal sequence is fed into the SOM. The error estimates are listed in a separate table, the output of the SOM is used to decide which one of

the error estimates is the best. The interference estimate is subtracted from the incoming signal before the classification. This kind of an interference cancellation can be combined with various equalizer structures. The cancellation can also be preceded by a DFE.
One possible architecture of the SOM based receiver is shown in Figure 69. In this structure, the DFE is used only for cancellation of intersymbol interference and multipath propagation. Other distortions are cancelled by the feedback loop, in which the error signal corresponding to the distortion, i.e. the interference and noise, is calculated and used in estimating the next value of the error for compensation.

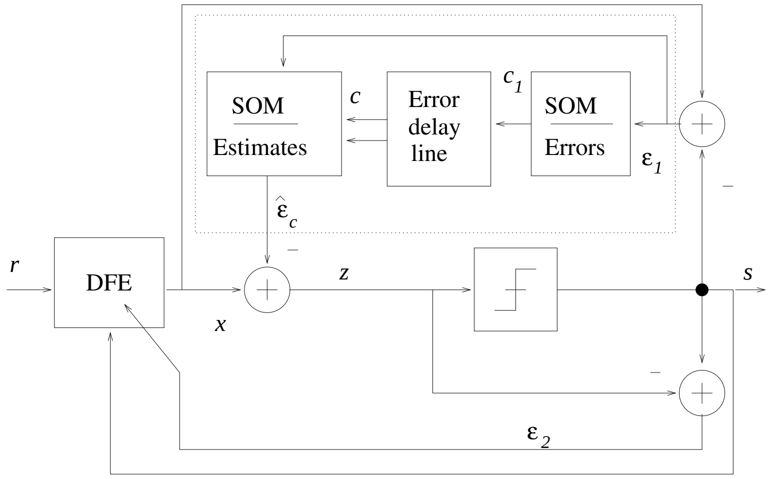
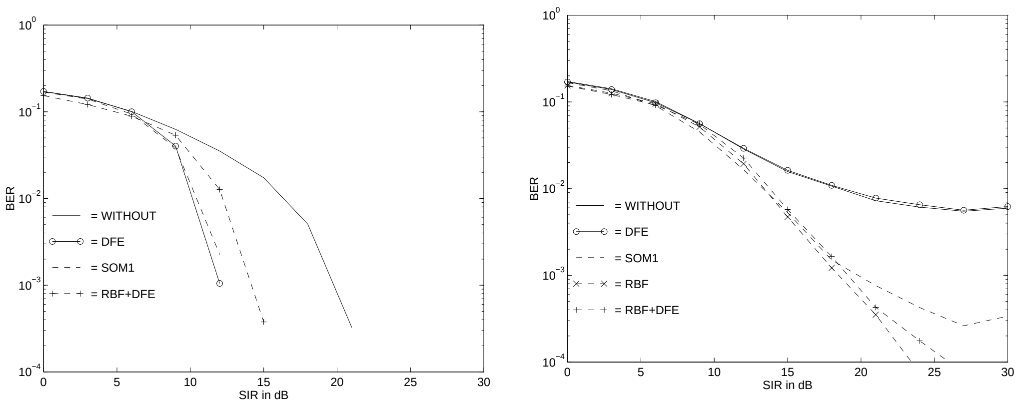


Figure 69: The sep-SOM receiver structure.

In computer simulations interference cancellation with various receiver structures utilizing the SOM algorithm have been investigated. We have concentrated on interference effects by using QAM modulation (16-QAM). Interference has been both from a Gaussian noise source and from a similar signal source as the desired signal. The latter with equally modulated signals is called co-channel interference (CCI). Some results of cancellation of the CCI are shown in Figure 70.



(a) Linear two-path channel (b) Nonlinear one-path channel

Figure 70: CCI cancellation

The BER vs. the signal to interference ratio (SIR) of the receivers have been studied. The SOM is compared to the DFE, the RBF network, and to the situation where no interference cancellation is performed. With no cancellation the samples are only classified to predefined states.

37.3 Conclusion

In this research, the Self-Organizing Map has been used in compensating nonlinear distortions and cancellation the Gaussian and co-channel interference. The simulation results have been derived using the QAM-modulation. The SOM based receivers are able to compensate nonlinear distortions, but their performance in interference cancellation is not so satisfactory.

References

[1] S. Haykin. *Neural Networks, a Comprehensive Foundation.* Macmillan, 1994.

[2] T. Kohonen. *Self-Organizing Maps.* Springer-Verlag, Berlin, 1995.

[3] T. Kohonen, K. Raivio, O. Simula, O. Ventä, and J. Henriksson. Combining linear equalization and self-organizing adaptation in dynamic discrete-signal detection. In *Proceedings of the International Joint Conference on Neural Networks*, volume 1, pages 223–228, San Diego, USA, June 1990.

[4] J. G. Proakis. *Digital Communications.* McGraw-Hill, 3 edition, 1995.

[5] K. Raivio, J. Henriksson, and O. Simula. Neural detection of QAM signal with strongly nonlinear receiver. *Neurocomputing*, 21:159–171, October 1998.

38 Adaptive Resource Management Methods in Telecommunications

Haitao Tang and Olli Simula

38.1 Resource Management Problems in Telecommunications

Adaptive resource management of telecommunication networks has become more and more important with the advent of Intelligent Networks, ATM, mobile communication networks, etc. With the increase of processing capacity alone, it may still not be possible to cope with the requirements, e.g., the Quality-of-Service (QoS) requirements to network. Thus, adaptive resource management methods are appreciated. In addition to their help to meet the requirements, the adaptive resource management systems have demonstrated their ability to increase system utilization, i.e., the more efficient use of system resources.

To solve the resource management problems, it is quite helpful to model the telecommunication networks with the ideas of server, client, and agent. The server, related to data networks, is a host in the network if it offers one or more services to the network, subnetworks, or applications in the networks. Likewise the network is the server for the hosts in the networks to get the needed transmission services. Those who receive the service(s) are called clients. For further dimensioning, sometimes, some components in the networks are called agents if they act between the server(s) and the clients. The agents pre-process the requests from the clients and route certain requests to certain server(s) for the required services; in this sense, the agents take both server and client roles. Therefore, the relation between a server and other components in the network can be modeled as the client-server or the client-agent-server. The server consists of certain elements, where some of the elements can take the manager role while others are simply the managed elements.

The server is usually a resource-sharing system (e.g., time-sharing system, memory-sharing system, or bandwidth-sharing system) which is the case in data networks. The server provides one or more services to the clients.

38.2 Adaptive Approaches to Resource Utilization

Generally, for the resource assignment, we have to consider two issues: the traffic adaptation and the adaptive resource allocation. The traffic adaptation is the first step of resource assignment, which understands the resource requirements for the services. The adaptive resource allocation is the final step of the resource assignment, which conducts the resource allocation for the requirements according to the traffic adaptation and server capacity (the effective resource of the server). When a server serves several services or service classes, the resource allocation for different service is correlated with each other, which may add difficulty to the adaptive resource allocation.

Because of the various traffic and server features, the solutions (if found) can be

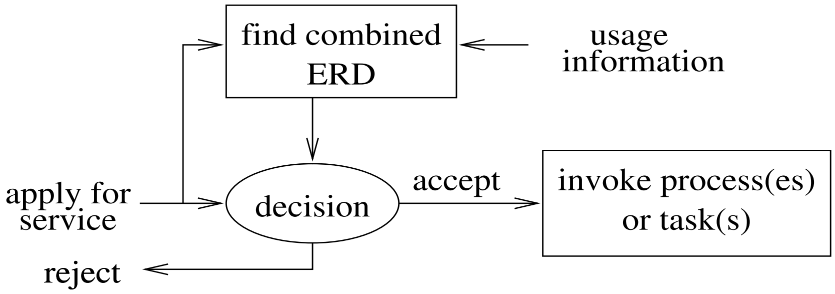


Figure 71: The principle of the admission control of an application for a service provided by the computer.

largely case-dependent or even heuristic. Thus, adaptive methods are used to solve the adaptation problems. They may help to decrease the case-dependency somehow, i.e., at least for the adaptive information processing part which is the heart of the resource assignment. Moreover, the adaptive methods may offer better solutions. There have been some studies which demonstrate the above advantages.

From 1995 to 1997, we proposed several methods for traffic adaptation, which were introduced in the 94-96 triennial report. They are not covered in this report. Since 1997, several other methods for traffic adaptation have been found. Furthermore, the corresponding resource allocation methods which include the resource reuse have been developed.

In the first approach, a mnemonic map (or age-boosting) is developed and used with the aging algorithm to adapt to the accessing environments of data pages and to increase the page access throughput of a database or other information systems. The scheme consists of the partially-weighted majority algorithm and the aging algorithm. These methods have been tested successfully by simulations. The results show: (a) the age-boosting scheme converges very fast; (b) the throughput improvement of the age-boosting scheme over the aging scheme is 8% - 45% and 22% on the average after the age-boosting scheme has converged. The integrated performance of the age-boosting scheme is much better than that of the aging scheme and is similar to that of the "fixed scheme".

The second approach is proposed for the systematic analyses of the effective capacities of a computer as a node in the network. The approach is developed through modeling a computer as a resources server, its components as resource objects, and its service tasks as resource-consuming processes with effective resource demands (ERDs). The service admission and the QoS management of the computer are then developed with the approach. A resource-consuming process on a resource object is the amount of resource units that the resource object should provide to the process in each time unit in order to achieve the given QoS requirement(s). The examples show that the approach is applicable. The principle of the service admission control is shown in Figure 71.

In the third approach, a combined flow control approach of IN is proposed as shown in Figure 72. A dynamic model is created to predict the SCP system state, where two methods are also proposed to adapt to a functional relationship in the model. The service rate for the flow from SMS (the internal flow control) is then decided according to the predicted system state and the guaranteed system response delay which is calculated through the analytical relationship among the system state, the system response delay, and the confidence level. The combined flow control is thus constructed by the internal flow control and ACG, which provides the needed performance.

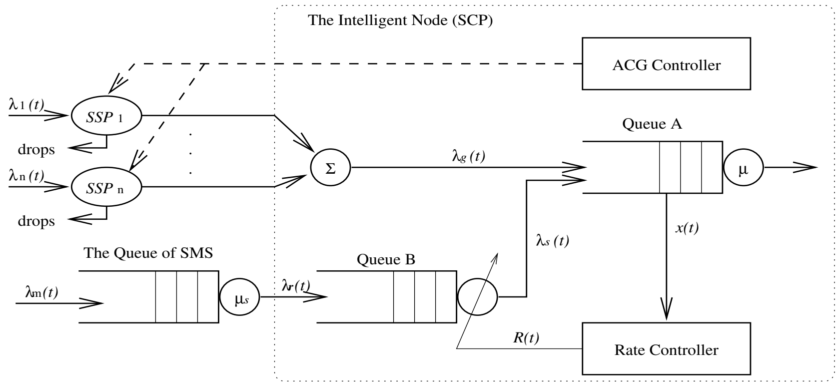


Figure 72: The queueing model for the major part of an Intelligent Network, where an SCP is connected with an SMS and n SSPs; the model is oriented to investigate the combined flow control.

As the fourth publication, the doctoral thesis of Haitao Tang summarizes some essential issues of the adaptive resource management and introduces the adaptive resource management approaches proposed by focusing mainly on the adaptive resource management of Intelligent Networks and computer-based end systems.

Publications

1. H. Tang, O. Simula, and K. Raatikainen, "Age-Boosting Page Replacement Scheme," International Conference on Telecommunications, Vol. 3, pp.1115-1120, 1997.

2. H. Tang and O. Simula, "The Effective Resource Demands of the Applications and Their Managements in a Computer," The Third Asia-Pacific Conference on Communications, Vol. 1, pp.267-271, 1997.

3. H. Tang and O. Simula, "Another Dimension of Flow Control for the Intelligent Node," International Conference on Telecommunications, Vol. 2, pp.425-429, 1998.

4. H. Tang, "Applying Adaptive Techniques and Operations Research Methods to the Resource Management in Telecommunications", ACTA POLYTECHNICA SCANDINAVICA, Ma 92, Espoo, 1998 (a publication of the doctoral thesis).

39 Subspace Techniques in CDMA Reception

Jyrki Joutsensalo and Juha Karhunen

The explosive growth in wireless communications, in conjunction with emerging new applications, has increased the demand for developing more efficient mobile radio systems. Analog mobile phone systems, such as NMT (Nordic Mobile Telephone) are called first generation systems. They were accompanied by digital second generation systems, for example GSM (Global System for Mobile communications). Those systems are primarily voice oriented; however, especially second generation systems also support low data rate services. Due to the high acceptance of cellular mobile radio systems, capacity limits have already emerged in highly populated areas. For capacity reasons and offering new services and system features, third generation mobile radio systems are under development. These systems emphasize the importance of coverage and access throughout the world. Trends towards multimedia applications and video transmission in mobile environment require high rate data transmission capability over radio interface. A potential practical system must provide reliable data transmission with very small bit error ratio. Third generation system should be able to adapt quickly to different user requirements and to build up solutions tailored to customers.

All these requirements increase the demand for more bandwith efficient multiple access schemes. There are several ways to allocate the frequency spectrum to users. The most widely known multiple access schemes are FDMA (Frequency Division Multiple Access; e.g. NMT) and TDMA (Time Division Multiple Access; e.g. GSM). Both methods rely on user partitioning in the time-frequency plane. The number of users that can be served is determined by the available frequency slots in FDMA (Figure 73 left), and by the available time slots in TDMA (Figure 73 middle). CDMA (Code Division Multiple Access) is considered as a promising solution for mobile communications. In CDMA all users are using the same frequency band at all times (Figure 73 right), as opposed to FDMA and TDMA. In CDMA, the separation of users is carried out by assigning each user with a unique code sequence (Figure 74).

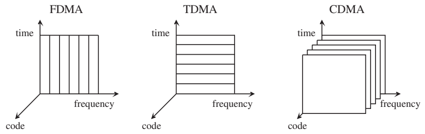


Figure 73: Different multiple access schemes.

Standard receiver structure for CDMA is simply a bank of matched filters (MF), kinds of keys, each matched to a particular user code. MF is optimum when only one mobile phone is in use, or when the codes of all the users appear to be orthogonal. In mobile environment orthogonality cannot in general be guaranteed because transmitted signal propagates through several paths (Figure 75).

In such a multipath environment standard receiver fails, especially when the different powers of different users received by the base station are very dissimilar due to their dissimilar distances to the base station. This is called near-far problem. Therefore,

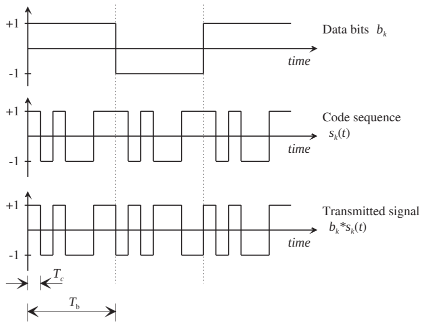


Figure 74: Illustration of the signal formulation.

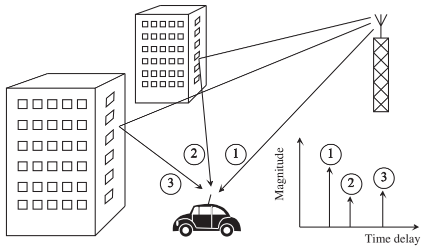


Figure 75: Urban multipath environment and channel impulse response.

more sophisticated receivers have been proposed. Common to all of these receivers is that they require knowledge of one or several parameters, such as propagation delay, carrier phase, and received power level. Frequently, the propagation delay estimate is a necessary prerequisite to the estimation of other parameters.
Conventional delay estimators are based on MF, correlator and delay-locked loop structures, but they suffer from the same performance degradation as standard receivers. More efficient methods have been introduced to overcome these problems. Theoretically optimal maximum likelihood method (MLM) requires multi-dimensional optimization, and thus is computationally too demanding in practice. Therefore, algorithms providing a trade-off between achieved performance and computational complexity are of primary interest. Subspace-based methods offer a potential solution for the delay estimation problem. Many of these methods are based on the projection matrix estimation via eigenanalysis. They use the knowledge of users' code sequences to form a parametric model for communication system. The observation space of the received signal is separated into the hyperplanes, called signal and noise subspaces (Figure 76).

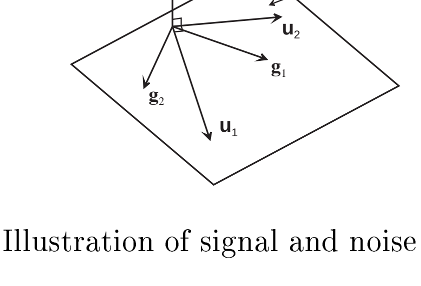


Figure 76: Illustration of signal and noise subspaces.

The power of the subspace methods lies in that under some mild conditions, signal and noise subspaces uniquely determine the unknown delays, even if the codes are not orthogonal. Code sequences are the only information needed on users. Moreover, multi-dimensional optimization problem is reduced to a series of one-dimensional problems, which decreases the computational requirements considerably. The best known subspace method called MUSIC (MUltiple SIgnal Classification) produces delay spectrum, from which one can estimate the delays by selecting those test delays producing the largest values to the spectrum (Figure 77).

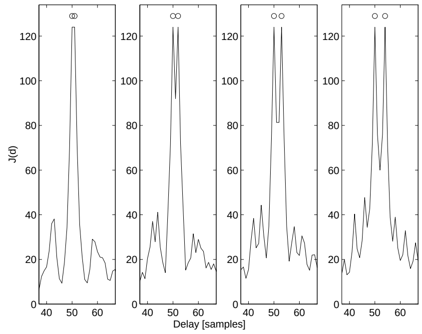


Figure 77: Resolvability of delays using MUSIC when spacing is 1, 2, 3, and 4 samples. Correct delays are indicated as circles.

This research project has been carried out in co-operation with Nokia Research Center. During the project, Mr. Petteri Luukkanen from Nokia Research Center finished his Master's Thesis *Subspace Delay Estimators for CDMA Systems* under the instruction of Dr. Jyrki Joutsensalo. The research project focused on the propagation delay estimation in asynchronous CDMA communication system. Performances of several novel and "classical" subspace methods have been evaluated in the multiuser multipath environment by Monte Carlo simulations. The estimators were compared to the conventional MF delay estimator. Subspace methods have earlier been studied in the Laboratory of Computer and Information Science in context with sinusoidal frequency estimation and array signal processing, leading to development of several new, computationally efficient variants of standard eigenvector-based MUSIC.
The main contributions of this project were:

- Development of hierarchic algorithms for implementing MLM as well as ACM-MUSIC type method. ACM-MUSIC has been introduced by the authors in 1990. Especially hierarchic MLM is computationally very attractive compared to the brute-force version of the MLM. The performance of the new algorithm HMLM (Hierarchic MLM) is based on the following fact: if the data can be modeled as linear combinations of some basis functions and noise, then the largest peak given by a simple linear transfrom based method, e.g. matched filter or sliding correlator, is usually close the true value. The value corresponding to the largest peak can be removed, and the next parameter can be estimated in similar manner. Under suitable conditions maximum likelihood estimator can be constructed hierarchically by estimating the parameters sequentially. This approach do not usually lead to the true values. However, it yields a very good initial estimate, so that one can finally perform a search

Algorithm	N	SNRs	4	3	2
HMLM	10	10,20,10,25	25 %	65 %	10 %
MUSIC	10	10,20,10,25	0 %	35 %	65 %
HMLM	10	20,30,20,35	45 %	50 %	5 %
MUSIC	10	20,30,20,35	5 %	45 %	50 %
HMLM	20	20,30,20,35	65 %	30 %	5 %
MUSIC	20	20,30,20,35	45 %	50 %	5 %
HMLM	50	0,10,0,15	65 %	20 %	15 %
MUSIC	50	0,10,0,15	0 %	55 %	45 %

Table 12: Comparing HMLM to MUSIC. Percentage of the number of correct delay estimates when two users and two paths exist. The total number of delays is four. N is number of observed symbols, and SNR:s are signal-to-noise ratios of different signals with respect to the noise.

Algorithm	SNR= 15 dB	SNR= 20 dB
MF	20.00%	31.00 %
MUSIC	28.00 %	69.25 %
ACM-MUSIC	99.50 %	100.00 %

Table 13: Comparison of different algorithms in nonstationary environments. The delays are changing, and users and paths are suddenly appearing or disappearing. Percentage of the number of correct delay estimates.

only in the vicinity of the estimated values, or use a gradient algorithm for fine tuning the estimates. Table 12 shows that HMLM works clearly better than MUSIC. The algorithm for hierarchic ACM-MUSIC is quite similar. See references [1,3,5,6].

- Modification of the ACM-MUSIC for tracking the changing delays. The performance of the algorithm is based on the assumption that the delays are changed sufficiently slowly so that one can detect the changes only near an "operating point", which is the point defined by previous estimates. In Table 13, the ACM-MUSIC has been compared to MF and MUSIC. See references [5,6].

- Introduction of criteria for estimating the model order. These criteria, on the contrary to the minimum description method based on the information theory, exploit efficiently the parametric form of the data, being still as simple as minimum description length (Figure 78). See references [2,3,5,6]. The new criteria have also been applied to sinusoidal frequency estimation and array signal processing [4].

- Introduction of simple methods for blind delay estimation and source separation in the mobile phone environment have led to the patent application.

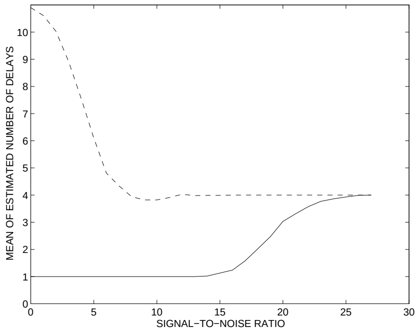


Figure 78: Comparing model order estimators in CDMA. The correct model order is four (two users with two paths). Solid: minimum description length; dashed: new method.

References

[1] J. Joutsensalo, J. Lilleberg, A. Hottinen, and J. Karhunen, "A Hierarchic Maximum Likelihood Method for Delay Estimation in CDMA". *Proc. IEEE 46th Vehicular Technology Conference* (VTC'96), Atlanta, Georgia, USA, April 28-May 1, 1996, vol. 1, pp. 188-192.

[2] J. Joutsensalo, "A Subspace Method for Model Order Estimation in CDMA". *Proc. IEEE Fourth International Symposium on Spread Spectrum Techniques and Applications* (ISSSTA'96), Mainz, Germany, September 22-25, 1996, vol. 2, pp. 688-692.

[3] J. Joutsensalo, J. Lilleberg, A. Hottinen, and J. Karhunen, "Hierarchic Method for CDMA Synchronization", *Proc. IEEE Nordic Signal Processing Symposium* (NORSIG'96), Espoo, Finland, September 24-27, 1996, pp. 17-20.

[4] J. Joutsensalo and A. Alastalo, "Linear Model Order Estimation by Signal Subspaces", *Proc. IEEE Nordic Signal Processing Symposium* (NORSIG'96), Espoo, Finland, September 24-27, 1996, pp. 319-322.

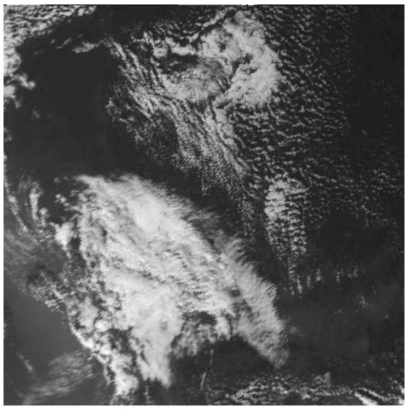
[5] J. Joutsensalo, J. Lilleberg, A. Hottinen, and J. Karhunen, "Subspace Algorithms for Synchronization and Tracking in CDMA". To be published in *Proc. IEEE Signal Processing Workshop on Signal Processing Advances in Wireless Communications*, (SPAWC'97), April 16-18, 1997, Paris, France.

[6] J. Joutsensalo, "Algorithms for Delay Estimation and Tracking in CDMA". To be published in *Proc. IEEE Int. Conf. on Communications* (ICC'97), June 8-12, 1997, Montreal, Canada.

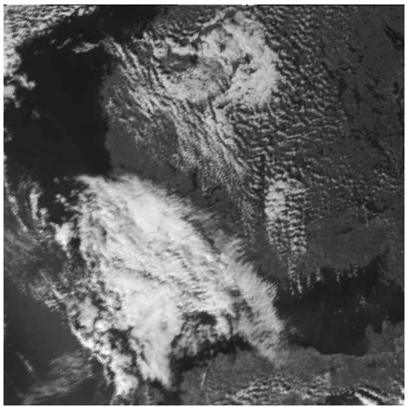
40 Satellite Image Analysis

**Jukka Iivarinen, Markus Peura, Olli Simula,
Kimmo Valkealahti, and Ari Visa**

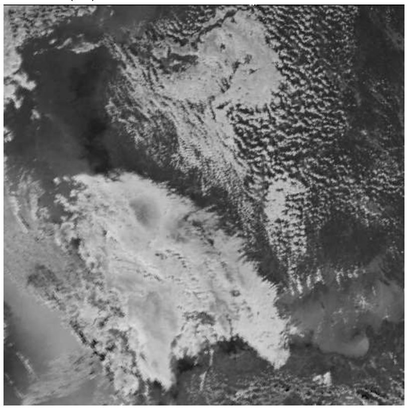
The Finnish Meteorological Institute (FMI) receives the high-resolution satellite images from weather satellites. The images are used for weather forecasting daily. The need for automatic methods to cloud detection was a main motivation for the present study. The automatic interpretation of satellite images has been studied in two projects in the Laboratory of Computer and Information Science in Helsinki University of Technology since 1991 [1-5]. In the first project the cloud classification was required over the Nordic Countries. In the second project the cloud cover and the cloud classification were required but only for some parts of Finland. The main interest has been the classification of clouds from satellite images by means of neural network methods.



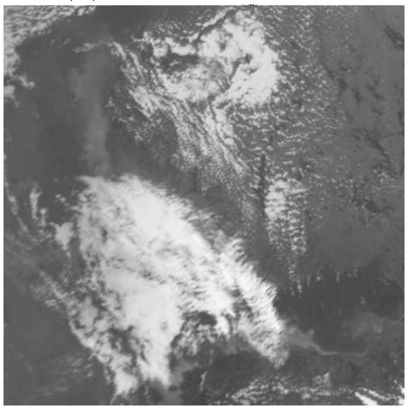
(a) Visible channel 1



(b) Visible channel 2



(c) Near-infrared channel 3



(d) Infrared channel 4

Figure 79: AVHRR channels of the NOAA-11 satellite image. The image was taken over southern Finland on 19th September, 1993, at 12:06 p.m. (GMT).

The applied satellite images are collected by the AVHRR on board the NOAA-10, the NOAA-11 and the NOAA-12 polar orbiting satellites. The AVHRR data consist of visible, near-infrared, and infrared channels (4 or 5 channels depending on satellite) (Figure 79). In Scandinavia the low level of daylight during the winter causes severe problems to the utilization of weather satellite images.

40.1 The Cloud Classifiers

The classification of a satellite image is performed in two phases. In the first phase the clouds are separated from the surface (referred as *cloud screening*), and in the second phase the cloudy regions are further classified into ten different cloud types (referred as *cloud classification*).

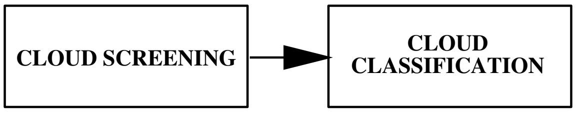


Figure 80: The classification is performed in two phases.

40.1.1 The First Classifier

The classification of clouds was based on the Self-Organizing Map (SOM) and on the Learning Vector Quantization (LVQ). A simple thresholding procedure was used in cloud screening. The selection of the Self-Organizing Map for the cloud classification in the present work was strongly motivated by the fact that no preclassified samples were needed for the initial training of the network. The feature maps were computed with hundreds of thousand unclassified feature vectors, obtained from tens of images acquired at different times of day and year. A small set of 260 preclassified samples was used only for the labeling and fine-tuning of the trained map.
The cloud classification was performed by extracting texture and spectral features from the information inside a gliding window. The window scanned all the bands of the satellite image at the same time. The extracted feature vector was fed to the classifying map. The classification result was obtained as a response from the best matching neuron. The classification result of the actual image point is the label of the best matching neuron. The procedure is shown in Figure 81.

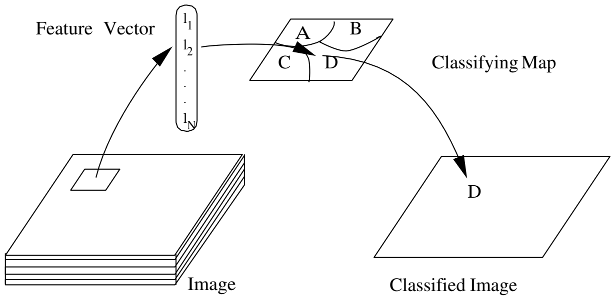


Figure 81: The cloud classification process.

40.1.2 The Second Classifier

When evaluating the first classifier there were situations where it clearly made false classifications. Because the classifier was taught with neural network methods, it was hard, sometimes impossible, to say what causes these false classifications. It is important for the finetuning of the classifier to know if the false classification is due to the classification method itself, or to something else, perhaps to a false classification of the preclassified sample.

This was the reason why a simplified classifier was used in the second approach. In the simplified classifier the codebooks are formed straight from the preclassified samples. This means that the feature vector of a preclassified sample is used as a codebook vector. The classifier consists of 32 codebooks. Each of the four seasons has four codebooks, a night and a day codebook for cloud screening and a night and a day codebook for cloud classification. The surface and cloud samples were collected by an experienced meteorologist. The samples were taken from the NOAA-11 satellite images between autumn 1991 and autumn 1993. The total number of samples is 1106.
In the cloud screening procedure a feature vector is extracted for each image pixel and compared with the codebook. The label of the best-matching codebook vector is presented as output. The classification of cloudy regions is then accomplished. A new feature vector is extracted and the classification is done as in the cloud screening procedure (Figure 82). The pooled form KNN algorithm ($K = 3$) with the Hamming distance is used to find the correct classifications in both procedures.

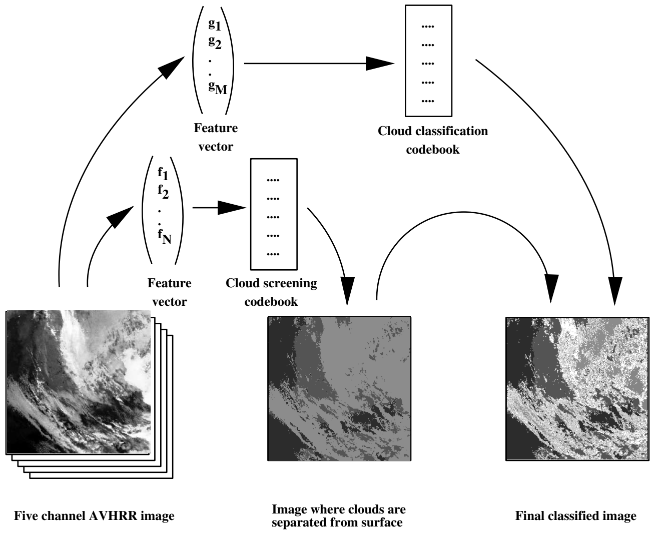
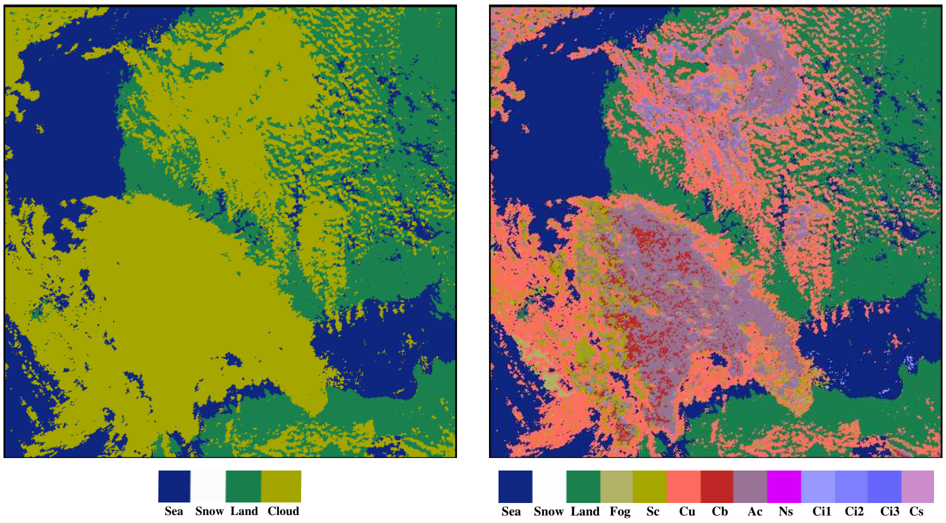


Figure 82: The classification procedure. Only cloudy regions are considered in cloud classification phase.

The classes used in cloud screening procedure were open sea, land, and cloud. In wintertime snow and ice were also classified (Figure 83(a)). The cloudy areas were further classified to ten cloud types which were cirrus over land/sea (Ci1), cirrus over low clouds (Ci2), cirrus over middle clouds (Ci3), cirrostratus (Cs), altostratus/altocumulus (Ac), stratus/stratocumulus (Sc), fog/stratus (Fog), cumulus (Cu), cumulonimbus (Cb), and nimbostratus (Ns) (Figure 83(b)).

40.2 Conclusions

Two versions of multispectral cloud classifiers were implemented to automate the processing of satellite images. The classifiers are automatic, and they can be adapted to changing situations by giving new examples. In the first approach the training of the classifier was done with the SOM and the LVQ algorithms. Neural networks offer a rapid way to get good results and to study the process. The use of neural



(a) Cloud screened image (b) Final classification

Figure 83: Classifications of the NOAA satellite image in Figure 1. In (a) is the classified image after cloud screening procedure, and in (b) is the final classification.

networks during the development process made it possible to reach the present stage in four years. However, in the final evaluation study a simplified classifier was used so that the false classifications could be traced and possible corrected.

The quality of the classifier has been verified with hundreds of images. In addition to the visual inspection, the automatic evaluation scheme is developed [1,3]. The comparisons with other published results show that the simplified classifier is working relatively well. The cloud classifier is in the evaluation use at the Finnish Meteorological Institute.

References

[1] J. Iivarinen, M. Peura, and A. Visa. Verification of a Multispectral Cloud Classifier. In *Proc. of the 9th Scandinavian Conference on Image Analysis*, vol. 1, pp. 591–599, Uppsala, Sweden, June 6–9 1995.

[2] O. Simula, A. Visa, and K. Valkealahti. Operational Cloud Classifier Based On the Topological Feature Map. In *Proc. of the International Conference on Artificial Neural Networks*, pp. 899–902, Amsterdam, September 13–16, 1993.

[3] A. Visa and J. Iivarinen. Evolution and Evaluation of a Trainable Cloud Classifier. *IEEE Trans. on Geoscience and Remote Sensing*, 35(5), pp. 1307–1315, 1997.

[4] A. Visa, K. Valkealahti, J. Iivarinen, and O. Simula. Experiences from Operational Cloud Classifier Based on Self-Organising Map. In *Applications of Artificial Neural Networks V*, Proc. SPIE 2243, pp. 484–495, 1994.

[5] A. Visa, K. Valkealahti, and O. Simula. Cloud Detection Based on Texture Segmentation by Neural Network Methods. In *Proc. of IEEE International Joint Conference on Neural Networks*, vol. 2, pp. 1001–1006, Singapore, November 18–21, 1991.

41 Land-Based Cloud Classification

Markus Peura and Ari Visa

41.1 Introduction

Classification of clouds has remained one of the few essential meteorological observations that have not yet been automatized. Typically, clouds have been detected by means of satellites; a recent study is reported by Visa et al. [1]. The scope in this study [2] was in *land-based* imaging of clouds, data being received by an all-sky imager. The research was initiated by Vaisala Oy, the company manufacturing detectors for weather observations. The emphasis was in developing recognition algorithms based on *visual appearance* of clouds. The hardware implementation was expected to apply visible and infrared domain. Another study involving land-based cloud classification is presented by Buch et al. [3].
In many problems of computer vision, the targets are distinct objects and the major challenge remains in optimizing image formation and in finding powerful features for classification. In our case, both the nature of the target (clouds) and the imaging method (land-based remote sensing) imply a challenging recognition task.

41.2 Basic idea

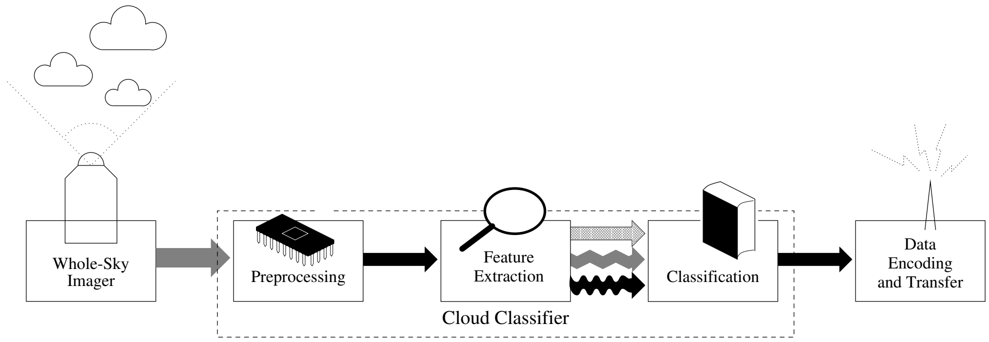


Figure 84: Proposed classification scheme

An outline of a device implementation is shown in Fig. 84. The aim for the classifier is to distinguish between thirteen target classes: ten cloud genera, fog, sun or clear sky. The elementary features are calculations between graylevels of neighboring pixels. These features are designed to indicate sharpness of cloud edges, fibrousness and specks of different size.
Typically, central parts of clouds belonging to different genera resemble each other by having similarly smooth appearance. In practice, this means that *classification at edges, being the most reliable, should be utilized in classifying the whole patch of a cloud.* That is, a cloud is seen as a fuzzy aggregate of segments containing some of the possible texture types of the respective genus. This principle was applied in the classification algorithm by propagating edge information to central areas of clouds.
In order to obtain a primary evaluation of the performance of the feature set, *Self-Organizing Map* [4] was applied. A small but comprehensive set of cloud images

was used as source data for the map shown in Fig. 85. The labels indicate classes of interest, cloud genera being subdivided to specks (S), edges (E), bulk (B) and gaps (G). The organization of the labels indicates some consistent clusters. The map verifies the intuitive assumption that different genera have visually similar details. Moreover, the intra-genus differences seem to be greater.
The smooth classes sun and sky occupy the center of the map, whereas specks, edges and other classes with pronounced graylevel variations are organized at the edges of the map. Precipitative clouds form an own cluster at the right edge of the map. The distinct cluster in the bottom refers to oddities in the applied imagery: branches of trees, street lamps etc.

41.3 Classification

For meteorological purposes, the required spatial resolution for the classification is much lower than the one of the source image. In addition, the subclasses of clouds (edges, specks, bulk, gaps) are of little interest and should be recombined to form integral specks of clouds. After pixelwise classification the image is divided to inner and outer areas in eight directions, resulting in total of 16 sectors. Each sector is labelled to the class having the largest amount of occurrences.
Two source images are shown in the top of Fig. 86. The final classification is shown in the bottom. Both images contain misclassified sectors. The errors are often logical: visual appearances of different cloud genera are known to be confusing in practice.
According to the experience obtained in this study, clouds, despite their physically complex nature, seem to be interpretable by means of image processing. Of course, performance of classification would be improved if altitude measurements were available. Nevertheless, this study can and should be seen as an indication of classification power when applying *visual information only*. It must be kept in mind that cloud classification is problematic also to human observers because no exact definitions exist for a cloud or cloud genera. Consequently, the results of the study can be considered promising.

References

[1] A. Visa, K. Valkealahti, J. Iivarinen and O. Simula. Experiences from operational cloud classifier based on self-organizing map. *SPIE Vol. 2243 Application of Artificial Neural Networks V*, vol. 2243, pp. 484-495, Orlando, Florida, April 1994.

[2] M. Peura, A. Visa, P. Kostamo. A new approach to land-based cloud classification. *Proceedings of 13th International Conference on Pattern Recognition*, pp. 143–147, Vienna, Austria, August 1996.

[3] K. A. Buch, Jr. and C.-H. Sun. Cloud classification using whole-sky imager data. *9th Symposium on Meteorological Observations and Instrumentation*, pp. 353–358, Charlotte, North Carolina, March 1995.

[4] T. Kohonen. The Self-Organizing Map. *Proceedings of the IEEE*, 78(9):1464-1480, 1990.

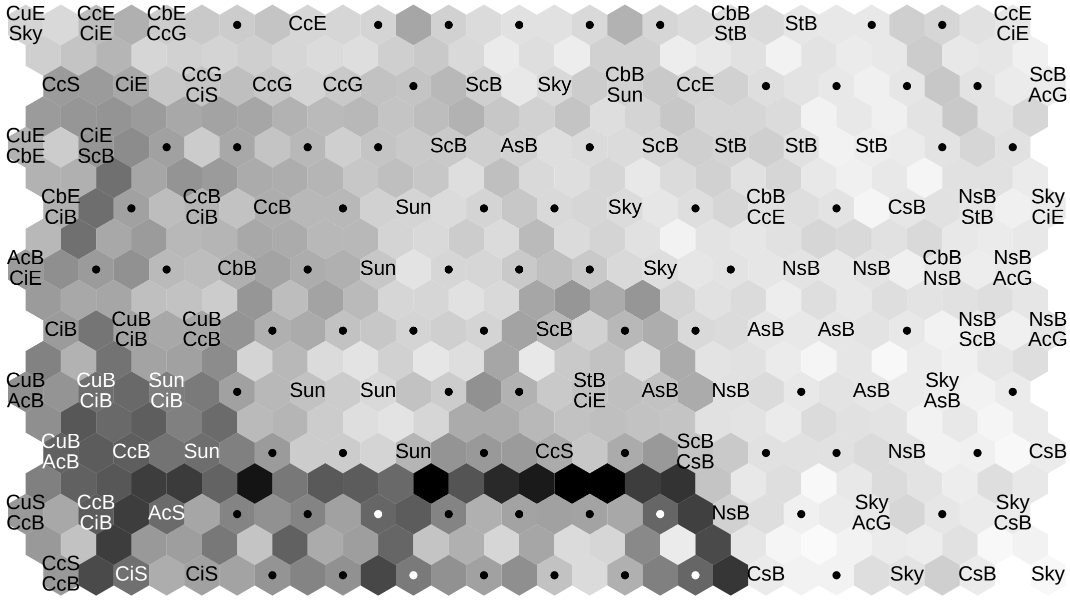


Figure 85: Class clusters on the Self-Organizing Map

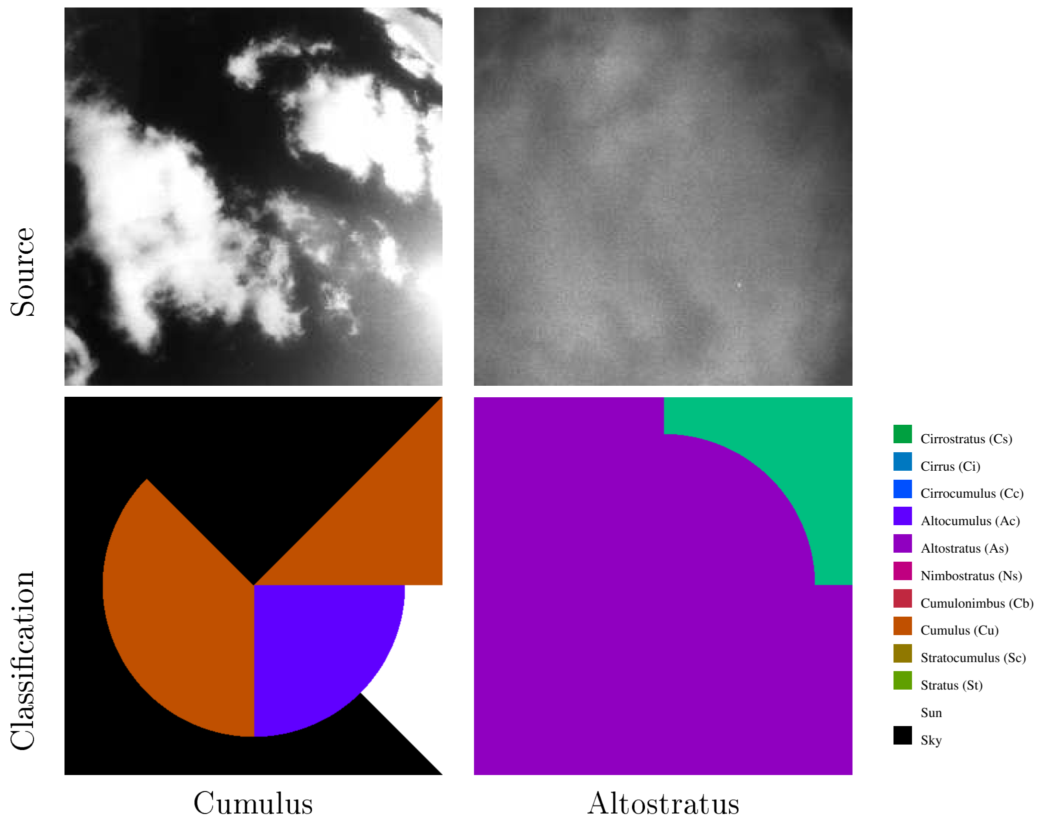


Figure 86: Raw all-sky images and classifications

42 Application of Statistical and Neural Classifiers to Recognition of Handwritten Digits

Erkki Oja, Jorma Laaksonen, and Lassi Tuura

Recently, many benchmark and comparison studies have been published on neural and statistical classifiers. One of the most extensive was the Statlog project [4] in which statistical methods, machine learning, and neural networks were compared using a large number of different data sets. As a general conclusion of that study, good statistical classifiers included the k-Nearest Neighbor (k-NN) rule, and good neural network classifiers included Learning Vector Quantization (LVQ) and Radial Basis Function (RBF) classifiers. One of the databases used in that study consisted of handwritten digits. The good performance of nearest neighbor classifiers for handwritten digits was also confirmed by Blue et al.[1], who however only compared the result to RBF and Multi-Layer Perceptron (MLP) neural classifiers. A kernel discriminant analysis (KDA) type method, the Probabilistic Neural Network (PNN), also did very well in that study.

In our study [3], we wanted to compare some classification methods developed within our research group to the standard classifiers. Special interest has been on the subspace methods of classification. The Averaged Learning Subspace Method (ALSM) [5] and some new modifications of it have been used in our experiments and compared against the classification error levels obtained with some other, more commonly used classification methods. As a case study, recognition of handwritten digits was used.

Off-line character recognition is one of the most popular practical applications of pattern recognition. During the last two decades, the research has mostly focused on handwritten characters. This is mainly due to the fact that while recognition of machine printed characters is considered a solved problem, the reliability achieved with handwritten text has not yet reached the level required in practical applications [2]. Handwritten character recognition has become a popular application area for neural network classifiers, which through their adaptive capabilities have often been able to achieve better reliability than classical statistical or knowledge-based structural recognition methods. For these reasons, we have also chosen to use handwritten digit data in our experiments.

The design of the feature extraction stage is an essential step in the development of a complete pattern recognition system. We have chosen to use simple statistical features obtained by applying Karhunen-Loève transformation on the normalized input images. Similar approach has been taken by many research groups in the Second Census OCR Conference [2] and also in the comparison [1]. Some exemplary images of handwritten digits are displayed in the leftmost column of Figure 42. The reconstruction of the original images from the calculated features is seen to get more accurate as the number of feature terms is increased.

Our evaluation study has been carried out by using systematic training set cross-validation in all classifier design. The final performance estimates are based on an independent testing set that has had no role in classifier construction, including the choice of optimal pattern vector dimension. The classification accuracies of

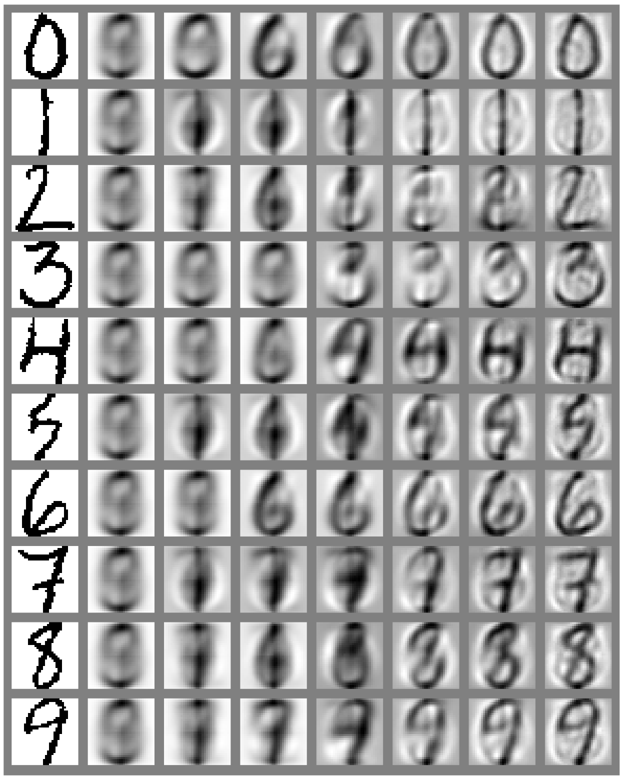


Figure 87: Some examples of handwritten digits and their reconstructions from increasing number of feature terms.

a subset of the classifiers tested are displayed in Table 14. While the case study does give some indication of the relative merits of the tested methods in the two particular applications studied, we want to emphasize that the results obtained by no means provide grounds for any objective ranking of these methods as alternative general classification schemes. The best classifier for a given task can be found by experimenting with different designs and basing the choice on criteria which, in addition to classification error, can include other issues such as computational complexity and feasibility of efficient hardware implementation.

classifier	**error-%**
KDA	3.5
MLP	3.5
LLR	2.8
k-NN	3.8
ALSM	3.2
committee	2.5

Table 14: Some examples of classification error levels obtained in the study.

As the last part of the evaluation, a committee comprising of three different classifiers was formed. By doing a majority vote of the outputs of the three member, the committee classifier was able to lower the error rate to a level below the accuracy of any of the classifiers involved. In the prototype handwritten digit recognition system developed for the current study, the option to reject a difficultly classifiable input image has also been implemented. As a general phenomenon in recognition of handwritten characters, it has been observed [2] that the rejection-error curve is linear in the $\rho \log \epsilon$-plane where ρ and ϵ denote overall rejection and misclassification rates, respectively. This feature has been confirmed by our experiments as seen in Figure 88.

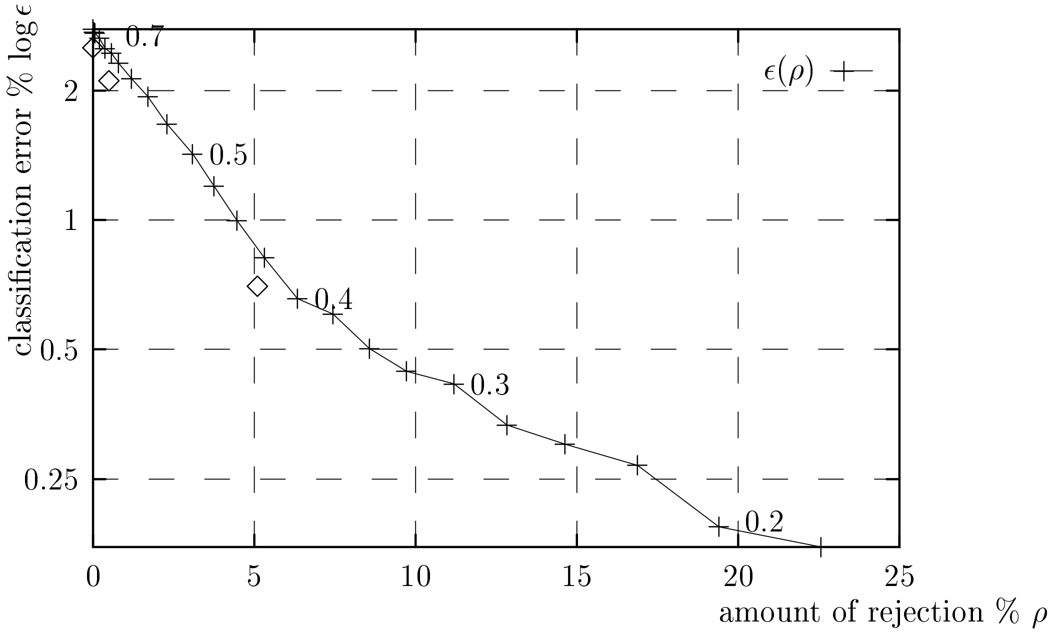


Figure 88: The rejection-error trade-off curve of the LLR classifier.

References

[1] J. L. Blue, G. T. Candela, P. J. Grother, R. Chellappa, and C. L. Wilson. Evaluation of pattern classifiers for fingerprint and OCR applications. *Pattern Recognition*, 27(4):485–501, 1994.

[2] J. Geist, R. A. Wilkinson, S. Janet, P. J. Grother, B. Hammond, N. W. Larsen, R. M. Klear, M. J. Matsko, C. J. C. Burges, R. Creecy, J. J. Hull, T. P. Vogl, and C. L. Wilson. The second census optical character recognition systems conference. Technical Report NISTIR 5452, National Institute of Standards and Technology, Aug. 1992.

[3] D. Michie, D. J. Spiegelhalter, and C. C. Taylor, editors. *Machine learning, neural and statistical classification.* Ellis Horwood Limited, 1994.

[4] E. Oja. *Subspace Methods of Pattern Recognition.* Research Studies Press Ltd., Letchworth, England, 1983.

43 Adaptive On-line Recognition of Handwritten Characters

Erkki Oja, Jorma Laaksonen, Vuokko Vuori, Matti Aksela, and Jarmo Hurri

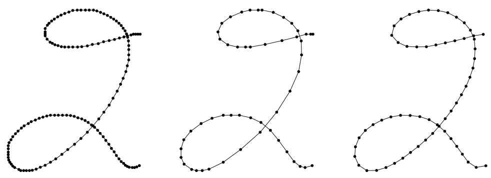
Automatic on-line recognition of handwritten text has been an on-going research problem for four decades. It has been gaining more interest lately due to the increasing popularity of hand-held computers, digital notebooks and advanced cellular phones. Traditionally, man-machine communication has been based on keyboard and pointing devices. These methods can be very inconvenient when the machine is only slightly bigger or same size as human palm. Therefore, handwriting recognition is a very attractive input method.

The most prominent problem in handwriting recognition is the vast variation in personal writing styles. There are also differences in one person's writing style depending on the context, mood of the writer and writing situation. The writing style may also evolve with time or practice. A recognition system should be insensitive to minor variations and still be able to distinguish different but sometimes very similar looking characters. Recognition systems should, at least in the beginning, be able to recognize many writing styles. Such multi-user systems usually have problems with recognition accuracy. One way to increase performance is adaptation, which means that the system learns its user's personal writing style.

The goal of the On-line Recognition of Handwritten Characters project is to develop adaptive methods for on-line recognition of handwritten characters. In this case, adaptation means that the system is able to learn new writing styles during its normal use. Due to the learning, the user can use his own natural style of writing instead of some constrained style. Our work concentrates on the recognition of isolated alphanumeric characters. The project is a part of the TEKES's technology programme Adaptive and Intelligent Systems Applications (AISA) and a subproject of the research project IMPRESS - Intelligent Methods for Processing and Exploration of Signal and Systems. The work is carried out in co-operation with Nokia Research Center.

The handwritten character recognition systems developed during this project are all based on template matching. They consist of a set of known characters (or prototypes), a similarity measure, and decision criteria. When a character is input to such a system, it is first preprocessed and normalized, then compared with all the prototypes, and finally classified according to its k nearest, or most similar, neighbors.

The preprocessing operations are very simple and they are used for altering the sampling of the characters. There were approximately 40 000 characters written by 46 subjects used in the experiments. The characters were collected with a system whose properties, such as sampling rate and resolution, are beyond the capabilities of the existing hand-held devices. Therefore, it was important to examine how sensitive the recognition methods are to the amount of data for each character. Prior to the matching, the characters are normalized by moving their centers, either mass or bounding box, onto each other. In addition, the characters can be rescaled



(a) (b) (c)

Figure 89: a) An original character and its two preprocessed versions when b) the sampling frequency is reduced to one third of the original frequency, and c) data points are sampled so that they are equidistant in space instead of time.

so that their bounding boxes are of equal size. In Figure 89, the original and two preprocessed versions of an example character are shown.
The prototype set is formed by clustering a large number of training samples and selecting one sample from each cluster to present all the samples in that cluster. The characters used for creating the prototype sets were written by different subjects than the character used for evaluating the performances of the recognizers. Therefore, all the experiments can be considered to be writer independent. The clustering algorithm applies the same similarity measure as the recognizer in the matching phase. Various similarity measures, all based on dynamic time warping (DTW) algorithm, have been suggested. DTW-algorithm enables nonlinear matching of curves or chain codes consisting of a variating number data points or items [7].
Adaptation of the recognition system is performed after each classification and it is based on the following ideas:

1. The prototype set is modified according to the recognition results of the new character samples. The prototype set is modified by by adding new prototypes, inactivating confusing prototypes, and reshaping existing prototypes with an algorithm based on Learning Vector Quantization (LVQ) [3]. These operations are carried out depending on how many of the k nearest prototypes belong to the correct class and their long term performances [8],[5],[6].

2. A set of classifiers is combined in a committee machine whose decision criteria are modified. The committee adaptation is based on the Dynamically Expanding Context (DEC) principle of Kohonen [1],[2]. The DEC principle adds new decision rules if the existing ones produce incorrect results. The new rules always strive to utilize more contextual information and are thus more specific than the old ones. In this case, the context is formed from the outputs of the committee members as is shown in Figure 90 [4].

Experiments performed with a recognition system which is able to adapt its prototype set have showed that a writer-independent classifier can be changed into a writer-dependent. Due to the adaptation, recognition accuracy high enough to be acceptable for a real-world application can be attained for most of the writers. An adaptation strategy which adds the input character into the prototype set if all the k nearest prototypes do not belong to the correct class was found to be the fastest

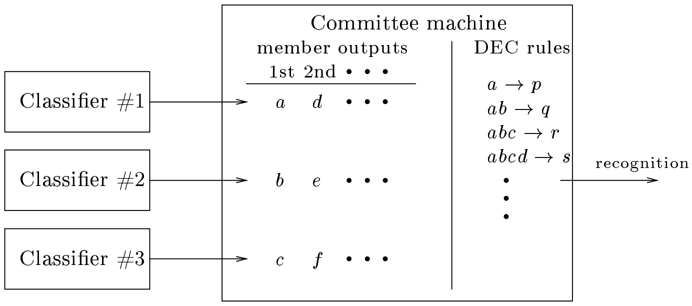


Figure 90: The basic setting of the DEC-based adaptive committee classifier.

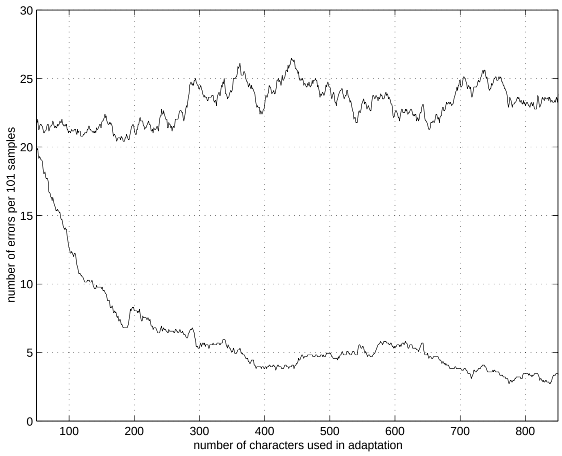


Figure 91: Evolution of the error rate during data collection in which the characters were recognized on-line. In the true data collection corresponding to the lower plot, input characters were added to the prototype set if one of the four nearest prototypes belonged to wrong class. The higher plot illustrates the nonadaptive simulation of the data collection. These result are obtained by averaging the result of eight writers.

way to decrease the error rate. However, the size on the prototype set increases considerably even if the recognition performance has ceased to improve.
Reshaping of the existing prototypes with a modified LVQ training rule nearly halves the error rate but is not sufficient when used alone as new writing styles fundamentally different from those represented by the initial prototypes cannot be learned. When these two adaptation strategies are combined so that new prototypes are added only if all the neighboring prototypes are incorrect and LVQ-learning is carried out otherwise, the growth of the prototype set is insignificant and the evolution of the recognition performance is nearly as good as with the pure adding strategy.
A part of the data was collected with a program that recognized the characters on-line and adapted itself to the writing style of the subject. The collected characters were both upper and lower case letters and digits. They were written in random order after the directions prompted by the program. The evolution of the error rate during the data collection and its nonadaptive simulation is illustrated in Figure 91 [8],[9].
A drawback of these adaptation strategies is that they all are supervised methods.

In practical applications, perfect supervision cannot be guaranteed. The recognition results are not necessarily correct even though they are accepted by the user, and, the user might change correctly recognized characters while correcting writing mistakes. In the future work, the recognition system's sensitivity to bad learning samples will be examined.

Several approaches to the DEC rules have been tested. Approaches for deciding the first result and order of the results taken by the system to form the context have been best classifier, majority voting, weighed majority voting, adjusting best classifier and adjusting majority voting. In the adjusting versions a measure of how well an individual classifier has performed has a direct contribution as to how decisive it is. As for the formation of the rules, examples of restrictions posed include requiring the output to belong to the context, predetermining the size of the context and various approaches to the situation where a new rule with the same context is created.

The use of the DEC rules has most notable effect when other adaptation is not used. Still, the use of the committee produces slight improvement in classification percentage when the members of the committee themselves are adaptive. Judging from the results available at this time, the most effective approach might be to at first have just the member classifiers adapt and begin using the DEC rules at a point when the error percentage of the individual classifiers has already reached a reasonably low level.

The main future goals of this project involve testing the sensitivity of the system in ambiguous situations and the development of a portable, hand-held testing and data gathering system.

References

[1] T. Kohonen. Dynamically Expanding Context, with applications to the correction of symbol strings in the recognition of continuous speech. In *Proceedings 8th International Conference on Pattern Recognition (8th ICPR)*, pages 1148–1151, Paris, France, October 1986.

[2] T. Kohonen. Dynamically expanding context. *Journal of Intelligent Systems*, 1(1):79–95, 1987.

[3] T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer-Verlag, 1997. Second Extended Edition.

[4] J. Laaksonen, M. Aksela, E. Oja, and J. Kangas. Dynamically Expanding Context as committee adaptation method in on-line recognition of handwritten latin characters. In *Proceedings of International Conference on Document Analysis and Recognition (ICDAR'99)*, 1999. Submitted.

[5] J. Laaksonen, J. Hurri, E. Oja, and J. Kangas. Comparison of adaptive strategies for on-line character recognition. In *Proceedings of International Conference on Artificial Neural Networks*, 1998.

[6] J. Laaksonen, J. Hurri, E. Oja, and J. Kangas. Experiments with a self-supervised adaptive classification strategy in on-line recognition of isolated handwritten latin characters. In *Proceedings of Sixth International Workshop on Frontiers in Handwriting Recognition*, pages 475–484, August 1998.

[7] D. Sankoff and J. B. Kruskal. *Time warps, string edits, and macromolecules: the theory and practice of sequence comparison.* Addison-Wesley, 1983.

[8] V. Vuori. Adaptation in on-line recognition of handwriting. Master's thesis, Helsinki University of Technology, 1999.

[9] V. Vuori, J. Laaksonen, E. Oja, and J. Kangas. On-line adaptation in recognition of handwritten alphanumeric characters. In *Proceedings of International Conference on Document Analysis and Recognition (ICDAR'99)*, 1999. Submitted.

44 Fault Analysis of Running Paper Web

Jukka Iivarinen and Ari Visa

On-line inspection is an essential part of modern web or sheet manufacturing. There are plenty of applications in different processes, e.g., in paper, nonwoven, plastic, metal, and plywood industries. The purpose of an inspection system is to detect and classify those defects which impair the quality of a product as compared to the requirements set by a user. The requirements mostly deal with the suitability of a product for the intended use of it. Certain defect types are so harmful that their presence in a product will make its further processing or converting difficult or impossible.
Typical characteristics of web manufacturing processes, when compared with other sheet or flat product manufacturing, are the large values of web width and production speed. Paper manufacturing is an extreme example of such demanding web processes. The web width of a modern paper machine may exceed 9 metres and its speed may reach 30 m/s. Such a machine makes about $3{\cdot}10^8\mathrm{mm}^2$ of paper each second, and all that production has to be inspected with a 100 % coverage. The inspection task is made more difficult by the fact that often the size of the smallest defects which have to be detected is smaller than a square millimeter.
Historically defect detection of web surfaces has been accomplished by hardware solutions for thresholding and matched filters. These techniques have made possible to detect only the most basic defect types. Detection of more complicated but critical defect types has remained unreliable. In this project more sophisticated methods have been developed. Use of texture and more complicated classifiers have become possible due to new sensor technology, increased calculation capability of computers and specialized hardware. Surface inspection has been studied in the Laboratory of Computer and Information Science at Helsinki University of Technology since 1995 [1-4]. The main interest has been the detection and classification of defects in a running paper web.

44.1 Overview of the Method

The proposed system model for web inspection has two phases: a segmentation phase (defect detection) and a classification phase (defect classification) (Figure 92). In the segmentation phase feature extraction is done and potential defect areas are marked. In the classification phase features describing the shape and internal structure of defects are extracted and defects are classified to different defect classes.

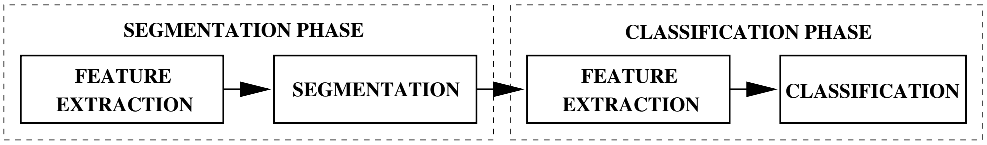


Figure 92: The proposed method consists of two phases.

The self-organizing maps (SOMs) are used in both phases. In defect detection a modified SOM, called the statistical SOM, is used to separate defects from a fault-free web. It is taught only with samples taken from a fault-free web. In defect classification the SOM is used to cluster unknown defects. It thus finds the classes (clusters) that are inherent to defect samples. These classes are then given an explanation (or a label) by hand.

44.1.1 Segmentation Phase

Images of many surfaces can be considered as stochastic textures, hence the co-occurrence matrices are used for the texture description. The co-occurrence matrix is reduced to a set of features to make calculation time and memory requirements smaller. The co-occurrence matrices are calculated locally within a small window that glides across the image.
The statistical self-organizing map is used to estimate the distribution of features extracted from faulty-free samples. Fault detection is based on the following idea: an unknown sample is classified to a defect if it differs enough from this estimated distribution. The segmentation scheme is depicted in Figure 93. An example base paper image and its segmentation is given in Figure 94.

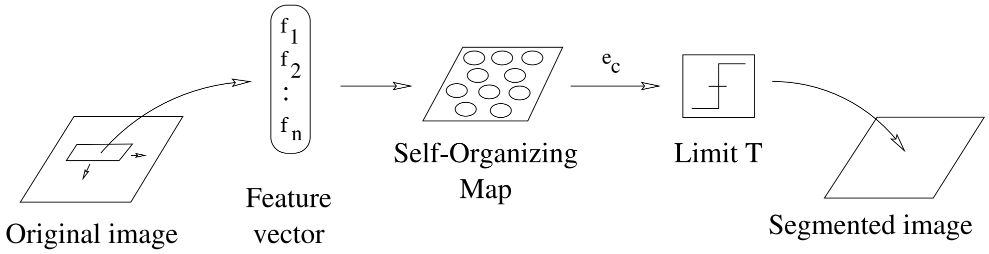


Figure 93: The segmentation scheme.

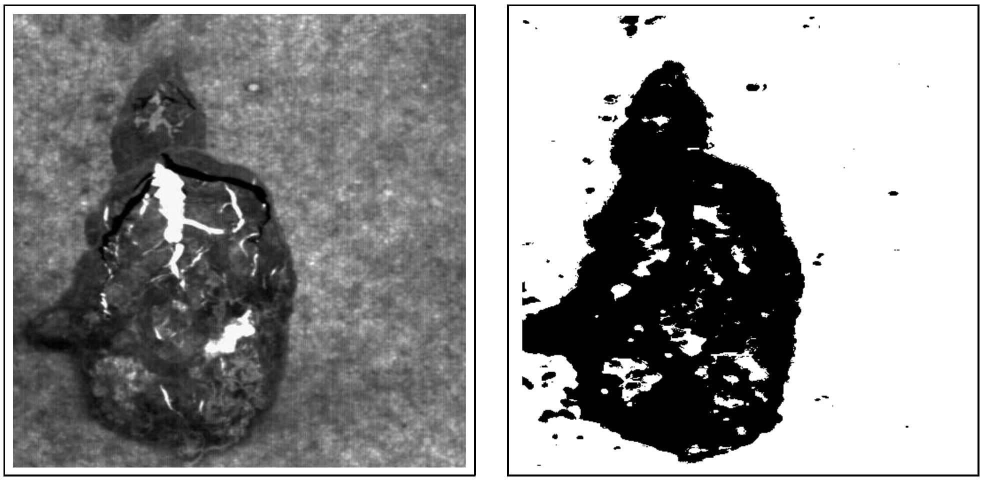


Figure 94: A base paper image and its segmentation. Defects are marked with black color.

44.1.2 Classification Phase

Due to the 2-dimensional nature of paper, most of its defects can be regarded as optical surface flaws which are detectable by a human eye. Therefore, classification

is commonly based on their visual appearance, namely on their shape and internal structure. Five simple shape descriptors are used to characterize the shape, and a gray level histogram and some co-occurrence matrix features are used for the internal structure.
The proposed defect classifier is depicted in Figure 95. There are three stages: a pre-processing stage, a classification stage, and a combiner stage. The classifier stage has three branches, one branch for each feature set. Each branch has a feature extraction unit and a classification unit. The outputs from the classification stage are combined in a combiner to produce the final classification.

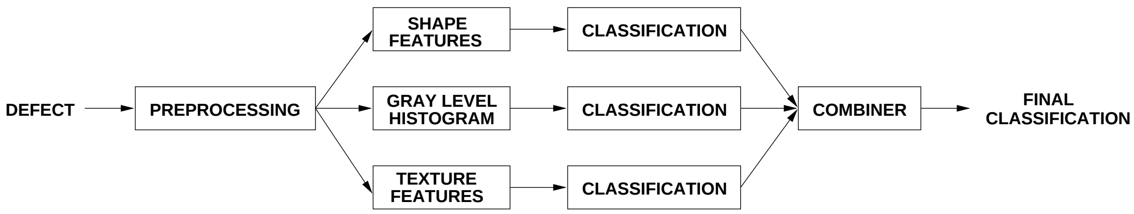


Figure 95: The defect classifier.

The codebooks are formed with the self-organizing map (SOM) algorithm. The advantage of using the SOM in codebook generation lies in the fact that the codebooks can be formed in an unsupervised way, simply by clustering unclassified training samples together, and then labeling the SOM units (or clusters) to represent different classes. In addition to manually given labels each SOM unit can be assigned a few representative defects. They act as examples of typical defects that belong to a SOM unit.
An example defect and its features are depicted in Figure 96(a). In Figure 96(b)-(d) are the classification results. Each column shows the best-matching codebook defect when using different features. In the bottom row are the numbers and the weight vectors of the best-matching SOM units. The first and the second row show one of the typical defects (with its feature vector) that belongs to the best-matching unit. The labels of the best-matching units are *elongated, smooth shape* (SSD), *light spot* (GLH), and *light spot* (TEX). The final classification is then *elongated, smooth, light spot*.

44.2 Conclusions

The two-stage approach offers advantages when considering real-time processes. Speed requirements of real-time defect detection and classification can be satisfied by splitting the needed procedure into two stages. The defect detection is a suitable part to hardware implementation. It is computing intensive. However, the adaptation is simple and concerns only the codebook vectors. More time can be spend on classifying found defects, because it can be assumed that defects are rare.
The proposed two-stage classification procedure is a general one and can be used in different classification problems. Reselection of features may be necessary to adapt the proposed classifier to work with different types of surfaces and defects. In defect detection the classifier is taught with examples of fault-free surface while in defect classification shape and internal structure characteristics of defects are

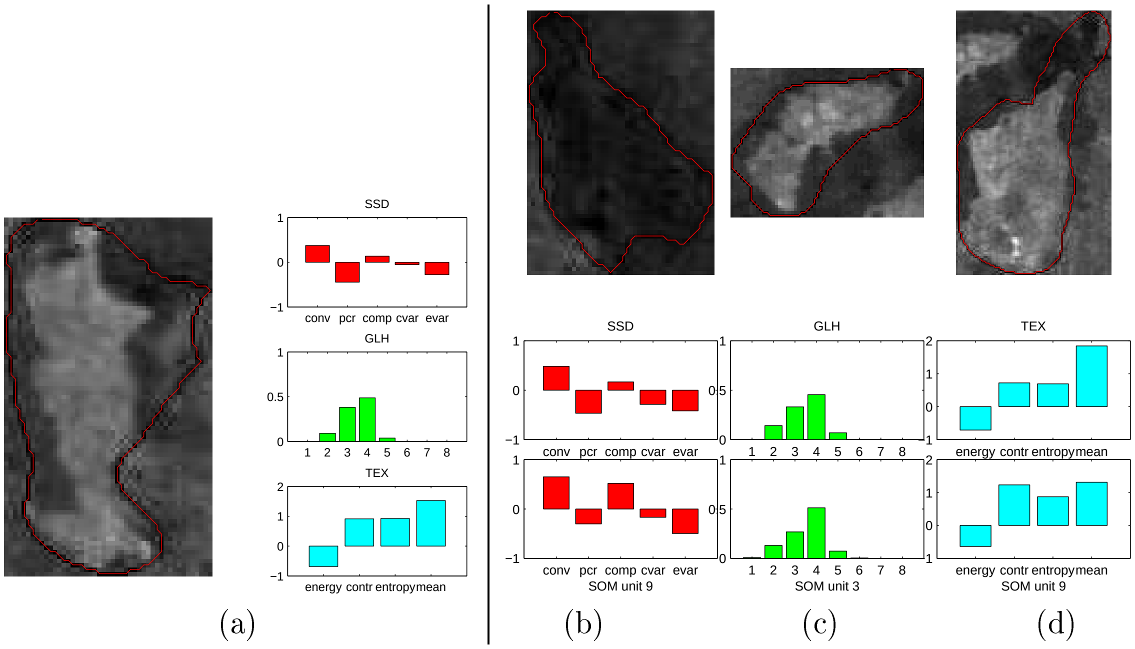


Figure 96: (a) A defect and its shape features (SSD), gray level histogram (GLH), and texture features (TEX). The best matching defects to the defect in (a) with respect to (b) the shape, (c) the gray level histogram, and (d) the texture features.

learned from examples. The self-organizing maps (SOMs) are used as classifiers. The defect classifier allows new, previously unknown, defects to be added to the codebook gradually during a long period. This is necessary since collecting samples of all possible defects is a time-consuming task and thus it is not reasonable to do it before initial training.

References

[1] J. Iivarinen and J. Rauhamaa. Surface inspection of web materials using the self-organizing map. In D. P. Casasent (Ed.), *Intelligent Robots and Computer Vision XVII: Algorithms, Techniques, and Active Vision*, Proc. SPIE 3522, pp. 96–103, 1998.

[2] J. Iivarinen, J. Rauhamaa, and A. Visa. An adaptive two-stage approach to classification of surface defects. In *Proc. of the 10th Scandinavian Conference on Image Analysis*, vol. I, pp. 317–322, Lappeenranta, Finland, June 9–11, 1997.

[3] J. Iivarinen, J. Rauhamaa, and A. Visa. Unsupervised segmentation of surface defects. In *Proc. of the 13th International Conference on Pattern Recognition*, vol. IV, pp. 356–360, Wien, Austria, August 25–30, 1996.

[4] J. Iivarinen and A. Visa. An adaptive texture and shape based defect classification. In *Proc. of the 14th International Conference on Pattern Recognition*, vol. I, pp. 117–122, Brisbane, Australia, August 16–20, 1998.

45 Analysis of Irregular and Hierarchical Visual Objects

Markus Peura

45.1 Introduction

Visual appearance of many natural objects is highly irregular and indefinite. The objects studied for example in biology, medical sciences, meteorology, and geomorphology have been challenges for computer vision. Examples of such objects are shown in Fig. 97. Dynamical behaviour and complex spatial hierachy are often additional difficulties in designing automated recognition schemes.

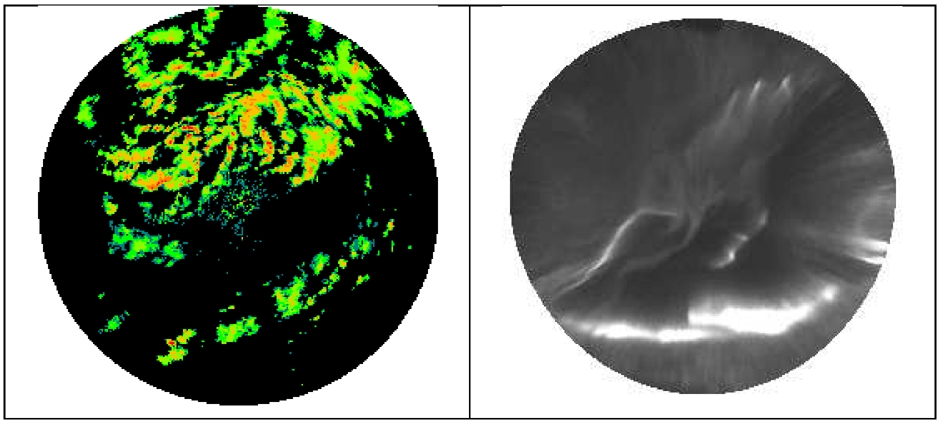


Figure 97: Examples of irregular natural images: precipitative clouds detected by a weather radar (left) and Northern lights captured by an all-sky camera (right).

45.2 Shape descriptors

A *shape descriptor* is an index, providing numerical information of a contour of an object [1]. Some descriptors are shown in Fig. 98. These descriptors were studied initially within the running paper project (Sec. 44). Convexity is a description the smoothness of an object, attaching penalty to every concavity (inlet) on its contour. Elongation is defined as the ratio of principal axes; the principal axes are the eigenvectors of the covariance matrix of a contour. Compactness is the ratio of a squared perimeter and an area. Variance and elliptic variance measure the shape difference from a circe and an ellipse, respectively.

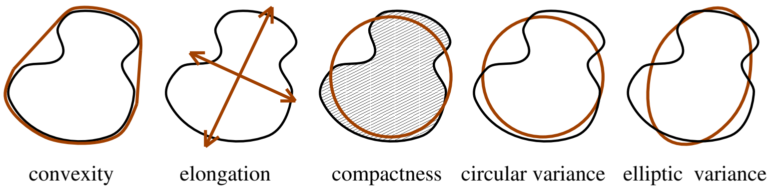


Figure 98: Shape descriptors.

45.3 Spatial hierarchy

As far as topology is considered, a natural way to perceive it is to regard intensities as altitude analogously to terrain elevation on a topographical map (Fig. 99 a and b). Finally, an attribute tree is obtained by attaching segment information (size, intensity, shape descriptions, etc.) to the obtained structure [2],[3].

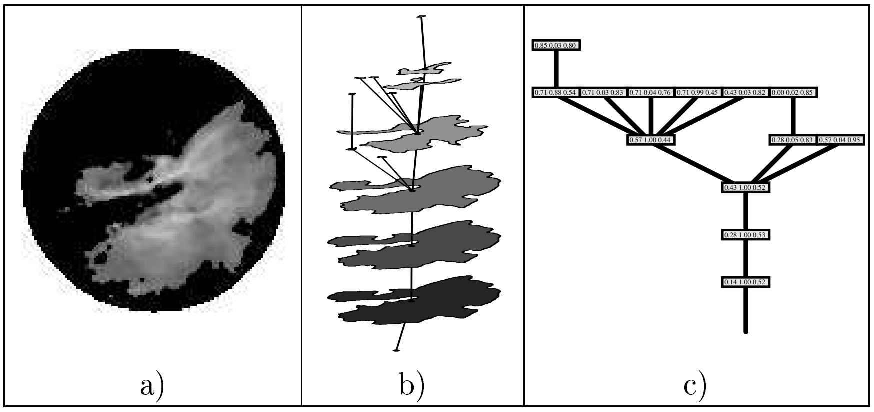


Figure 99: Original image (a), its topology (b) and attribute tree (c).

45.4 Indexing and matching attribute trees

As to recognizing and classifying obtained attribute trees, graph matching is a rigorous but rather elaborous technique. In this study, new fast techniques for indexing, matching, and generalizing unordered attribute trees have been developed. The proposed matching scheme is based on dividing the tree recursively into subtrees. The subtrees are matched according to indices, which have been calculated in advance using linear updating rules. In other words, exhaustive matching of subtrees is replaced by matching points in space. The overall computation time is linear.
The descriptors used in this study are height, node count, centroid, and branching variance, which have straightforward real-world analogies. The height of a tree is the length of the longest branch starting from the root. The descendant count can be thought as the mass of a tree. The centroid is an indicator of the vertical distribution of the mass. The branching variance measures structural irregularity. When matching attribute trees, these descriptors can be readily generalized to the attributes. One step of the matching scheme is illustrated in Fig. 100, where height and node count have been outlined as the height and width of the rectangles, respectively.

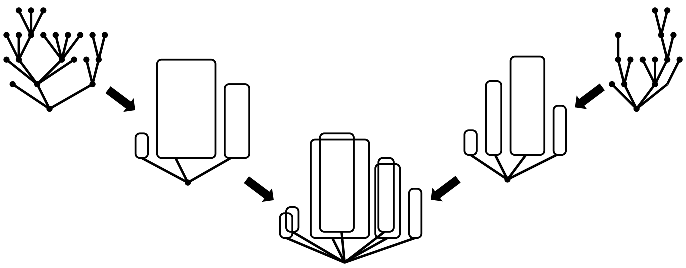


Figure 100: Illustration of the index-based heuristic matching.

45.5 The self-organizing map of attribute trees

After matching two trees, the resulting tree can be *weighted*. Weighted instatiations of the matched trees are essentially interpolations and imply direct applicability in learning systems involving prototype generation trough averaging. The *self-organizing map of trees* [4] is an extension of the standard self-organizing map (using vectors); the key issues is the revised definitions for a distance metric and adjusting. A map trained with 1115 weather radar images is shown in Fig. 101. The applied attribute vectors have three elements: intensity (red), are (green) and elongation (blue).

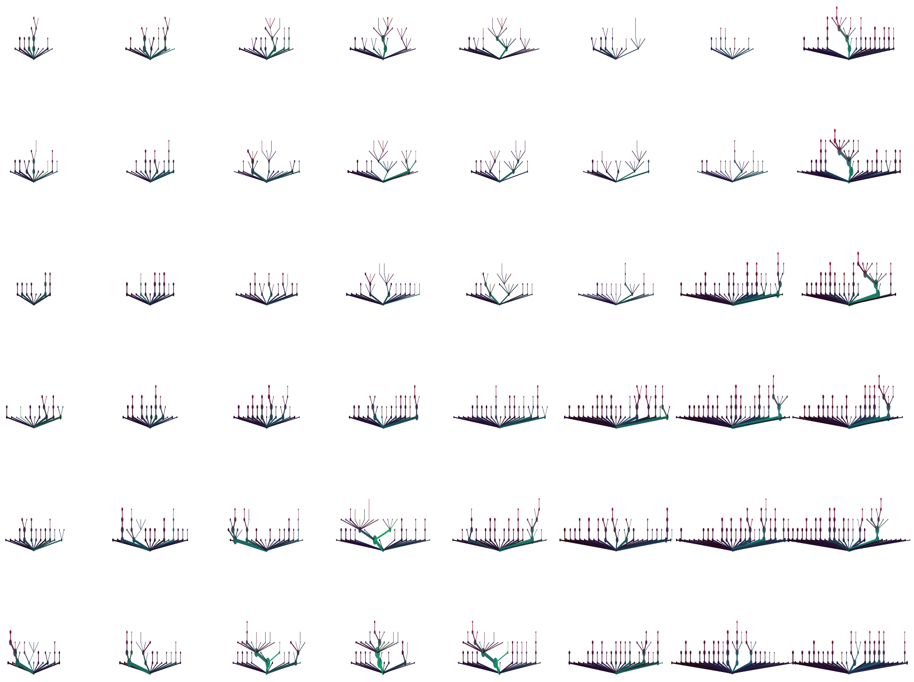


Figure 101: A self-organizing map of attribute trees.

References

[1] Iivarinen, J., Peura, M., Särelä, J. and Visa, A.: "Comparison of Combined Shape Descriptors for Irregular Objects". BMVC'97, 8th British Machine Vision Conference, Essex, Great Britain, September, 1997.

[2] Peura, M.: "A Statistical Classification Method for Hierarchical Irregular Objects". Image Analysis and Processing, Vol. 1, pp. 604-611, Alberto del Bimbo (Ed.), Lecture Notes in Computer Science, Springer Verlag, September 1997.

[3] Peura, M., Koistinen, J., and King, R.: "Visual modelling of radar images". COST-75 Advanced weather radar systems - International seminar, pp. 307-317. Locarno, Switzerland, March 1998. European Commission.

[4] Peura, M.: "The Self-Organizing Map of Trees". *Neural Processing Letters*, 8(2), pp. 155-162, October 1998. Kluwer Academic Press.

46 Texture Classification with Reduced Multidimensional Histograms

Erkki Oja and Kimmo Valkealahti

Texture refers to visual or tactile surface characteristics which are described by such terms as smoothness, roughness, regularity, uniformity, and granularity. Texture plays an important role in the visual perception of objects. Figure 102 shows six surface images, which are more or less immediately perceived as distinct textures. The luminance distributions of the textures are equalized so that their one-dimensional gray-level distributions are equal, i.e., the number of times each gray level occurs is the same in all textures. As in vision, computerized discrimination of texture images with identical gray-level distributions is based on spatial relationships among pixels.

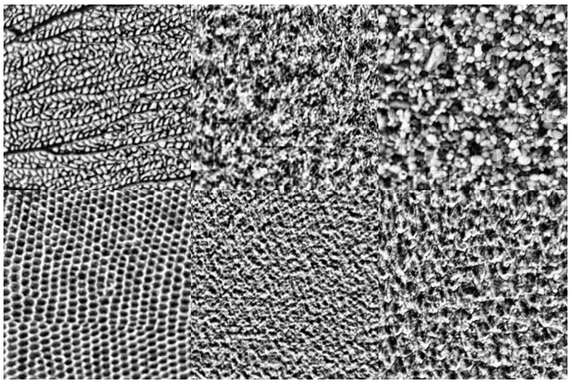


Figure 102: Textures with identical luminance distributions.

In 1962, vision researcher Bela Julesz proposed his famous conjecture that texture pairs with identical two-dimensional gray-level distributions, i.e., joint distributions of values of two pixels with any spatial separation, are not visually discernible [1]. Julesz's subsequent studies provided counterexamples to this conjecture [2], such as the three textures in Fig. 103 whose three-dimensional black-and-white distributions are the same for all combinations of three pixels. A recent study of Purpura et al. [6] showed that the primary visual cortex, the first stage of cortical processing, extracts multidimensional contextual dependencies in texture images.

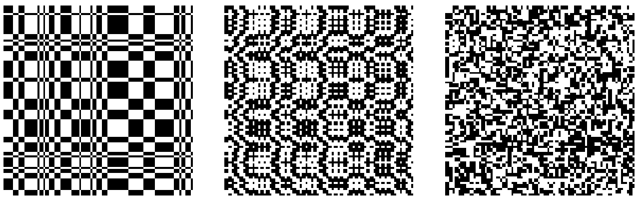
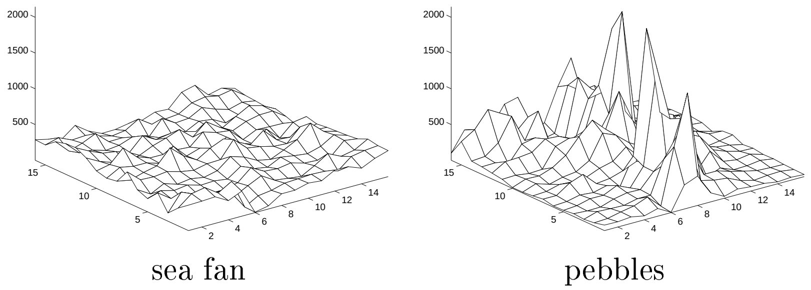


Figure 103: Black-and-white textures with identical three-dimensional distributions.

Julesz's early conjecture is still frequently cited to support the use of only two-dimensional co-occurrence statistics for machine-based discrimination of natural textures. Observations on vision made us suggest that the analysis of multidimensional dependencies benefits computerized texture discrimination. The analysis of multidimensional dependencies requires the use of multidimensional histograms, which is complicated by rapid expansion of histograms with increasing number of pixels and quantization levels. Histogram expansion, without increase in sample size, leads to decrease in bin frequency and consequently, to decrease in the reliability of probability distribution approximations. At small sample sizes typical of texture analysis, large multidimensional histograms must therefore be reduced by combining adjacent bins. At present, there are no standard reduction methods for this purpose. Our study has three main goals: comparison of multidimensional statistics with conventional methods in texture classification, development of methods for reduction of multidimensional histograms, and selection of multidimensional co-occurrence features and parameters of the classifier with respect to their performance in texture classification.

Our first experiments with both unreduced and reduced multidimensional histograms showed that their performance may exceed that of two-dimensional histograms [4,5]. Effective reduction of multidimensional histograms, leading to decrease in the classification error rate, was obtained using vector quantization with the self-organizing map [3]. In this reduction method, the co-occurrence vectors of pixel values in a predefined spatial arrangement are quantized using the reference vectors of a trained two-dimensional self-organizing map. The reference vectors of a trained map are adapted to the high-density regions in the co-occurrence distribution of the samples used for the training of the map. A reduced texture histogram is obtained when the reference vectors of a trained map are used as histogram bins to collect a two-dimensional map histogram of quantized vectors from a texture sample. Texture classification is then carried out by matching sample histograms with precomputed texture model histograms. With this approach we showed, both for monochrome and color textures, that codebooks trained with the self-organizing map algorithm provided significantly higher classification accuracy than two- and multidimensional unreduced histograms.



sea fan pebbles

Figure 104: Seven-dimensional histograms reduced by the self-organizing map.

The self-organizing map preserves similarity relationships within the data so that reference vectors near each other resemble each other. Thus, a histogram reduced with the map is easy to visualize. This is demonstrated by Fig. 104 showing the map histograms of upper left and right textures in Fig. 102, sea fan and pebbles. The reduced histograms represent seven-dimensional distributions of high-pass-filtered co-occurrence vectors whose components were sampled within 3-by-3-pixel windows. The local uniformity of the pebble texture is reflected in the high peaks in the histogram. In the sea fan texture, the edges in different directions are represented by separate histogram bins which results in a flat histogram.

A method for the selection of co-occurrence features and histogram size was also developed. To minimize the expected quantization error, vector quantizer algorithms tend to concentrate reference vectors along the directions with largest variance. This arrangement of a limited number of reference vectors may not be the best for classification. Our studies showed that whitening of co-occurrence distribution may improve classification accuracy. The complexity of a texture classifier is determined by the number and dimension of the reference vectors. The classification accuracy decreases if the number of parameters becomes too high or too low. In our study, the trade-off was found using a genetic algorithm to minimize the classification error rate. The most recent results of this research appeared in [7], as well as in the D.Sc. Thesis of Mr. Valkealahti (1998).

In conclusion, our studies suggest that texture classification is improved by increasing dimensionality of co-occurrence features, that the self-organizing map is suitable for the reduction of multidimensional co-occurrence histograms, and that the classification accuracies can be substantially improved by optimization of features used in construction of features vectors and by optimization of classifier parameters.

References

[1] B. Julesz. Visual pattern discrimination. *IRE Transactions on Information Theory*, IT-8(2):84–92, February 1962.

[2] B. Julesz, E. N. Gilbert, and J. Victor. Visual discrimination of textures with identical third-order statistics. *Biological Cybernetics*, 31:137–140, 1978.

[3] T. Kohonen. *Self-Organizing Maps.* Springer-Verlag, Berlin, 1995.

[4] E. Oja and K. Valkealahti. Compressing higher-order co-occurrences for texture analysis using the self-organizing map. In *Proceedings of the IEEE International Conference on Neural Networks*, volume 2, pages 1160–1164, Perth, Western Australia, November 27 – December 1 1995.

[5] E. Oja and K. Valkealahti. Co-occurrence map: quantizing multidimensional texture histograms. *Pattern Recognition Letters*, 17(7):723–730, June 1996.

[6] K. P. Purpura, J. D. Victor, and E. Katz. Striate cortex extracts higher-order spatial correlations from visual textures. *Proceedings of the National Academy of Sciences USA*, 91(18):8482–8486, August 1994.

[7] Valkealahti, K. and Oja, E. Reduced multidimensional co-occurrence histograms in texture classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence 20*: 90 - 94, 1998.

47 PicSOM: Self-Organizing Maps for Content-Based Image Retrieval

Erkki Oja, Jorma Laaksonen, Markus Koskela, and Sami Brandt

Content-based image retrieval from unannotated image databases has been an object for ongoing research for a long period. Digital image and video libraries are becoming more common and growing in size as more visual information is produced at a rapidly increasing rate. The technologies needed for retrieving and browsing this accumulating amount of information are still, however, quite immature and inadequate for many practical applications. Many projects have been started in recent years to research and develop systems for content-based image retrieval, of which best-known being the Query By Image Content (QBIC) [1] developed at the IBM Almaden Research Center.

We have started to develop methods to utilize the strong self-organizing power of the Self-Organizing Map (SOM) [2] in unsupervised statistical data analysis to facilitate content-based retrieval from large image databases [4,5]. Our experimental image retrieval system is named PicSOM, bearing similarity to the WEBSOM document browsing and exploration tool. PicSOM uses a World Wide Web browser as the user interface and a hierarchical version of the SOM algorithm called Tree Structured Self-Organizing Map (TS-SOM) [3] as the image similarity scoring method. The TS-SOM is a tree-structured vector quantization algorithm that uses two-dimensional

Figure 105: WWW-based user interface of PicSOM. The user has previously selected five aircraft images. The system is displaying the user ten new images to select of.

SOMs at each of its hierarchical levels. The implementation of PicSOM is based on a general framework in which the interfaces of co-operating modules are defined.

Figure 105 shows a screenshot of the current web-based PicSOM user interface, which can be found at *http://www.cis.hut.fi/picsom/*. A notable feature in PicSOM is its ability to use multiple reference images, while most other current systems are based on using a single reference image.

The retrieval approach in PicSOM is based on relevance feedback [6] adopted from traditional information retrieval techniques. In relevance feedback, implicit information of the human-computer interaction during previous queries is used to refine the response on subsequent rounds. The image queries are thus iteratively refined during the retrieval process, as the system exposes more images to the user and tries to adapt to the user's preferences regarding the similarity of images.

The system may use one or several types of statistical features for image querying. Separate feature vectors can be formed for describing, for example, the color content, various textures, and objects of the images. A separate TS-SOM is then constructed for each feature vector set and these maps are used in parallel to calculate the best-scoring similarity results. The feature selection is not restricted in any way and new features can be added to the system later on.

Combining the results from several maps can be done in a number of ways. A simple method would be to ask the user to enter weights for different maps and then calculate a weighted average. This, however, requires the user to give information which she normally does not have, as it is a difficult task to give low-level features such weights which would coincide with human's perception of images. Therefore, a better solution is to apply the relevance feedback approach, in which the results of multiple maps are combined automatically, using the implicit information from the user's responses during the query.

The rationale behind our approach is as follows: If the images selected by the user map close to each other on a certain TS-SOM map, it seems that the corresponding feature performs well on the present query and the relative weight of its opinion should be increased. This can be implemented by marking the images the user has seen on the maps. The units are given positive and negative values depending whether she has selected or rejected the corresponding images. The mutual relations of positively-marked units residing near each other can then be enhanced by convolving the maps with a simple low-pass filtering mask. As a result, areas with many positively marked images spread the positive response to their neighboring map units. The images associated with these units are then good candidates for next images to be shown to the user, if they have not been shown already.

Figure 106 shows a set of convolved feature maps during a query. The three images on the left represent three map levels on the TS-SOM for a RGB color feature, whereas the convolutions on the right are calculated on a texture map. The sizes of the SOM layers are 4×4, 16×16, and 64×64, from top to bottom.

The research will continue along several lines: To increase PicSOM's retrieval performance, we need to add better feature representations. These will include color histograms, color layout descriptions, shape features, and some more sophisticated texture models. As the PicSOM architecture is designed to be modular and expandable, adding new features is straightforward. We are also developing quantitative measures to compare the performance of different features and of PicSOM with

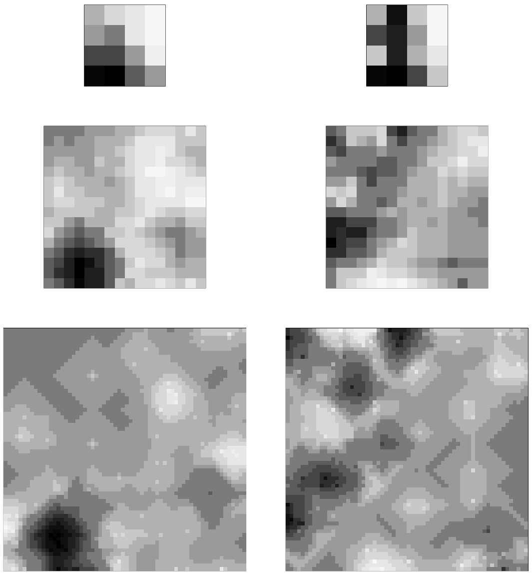


Figure 106: An example of convolved TS-SOMs for color (left) and texture (right) features. Black corresponds to positive and white to negative convolved values.

that of other content-based image retrieval systems. Quantitative measures of the image retrieval performance are, however, problematic due to human subjectivity. Generally, there exists no definite right answer to an image query as each user has individual expectations. Furthermore, to study our method's applicability on a larger scale we shall need larger image databases. A vast collection of images is available on the Internet, and we have preliminary plans to use PicSOM as an image search engine for the World Wide Web.

References

[1] M. Flickner, H. Sawhney, W. Niblack, et al. Query by image and video content: The QBIC system. *IEEE Computer*, pages 23–31, September 1995.

[2] T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer-Verlag, 1997. Second Extended Edition.

[3] P. Koikkalainen. Progress with the tree-structured self-organizing map. In A. G. Cohn, editor, *11th European Conference on Artificial Intelligence*. European Committee for Artificial Intelligence (ECCAI), John Wiley & Sons, Ltd., August 1994.

[4] J. Laaksonen, M. Koskela, and E. Oja. Content-based image retrieval using self-organizing maps. In *Third International Conference on Visual Information Systems*, Amsterdam, The Netherlands, June 1999.

[5] J. Laaksonen, M. Koskela, and E. Oja. PicSOM - a framework for content-based image database retrieval using self-organizing maps. In *11th Scandinavian Conference on Image Analysis*, Kangerlussuaq, Greenland, June 1999.

[6] Y. Rui, T. S. Huang, and S. Mehrotra. Relevance feedback techniques in interactive content-based image retrieval. In I. K. Sethi and R. J. Jain, editors, *Storage and Retrieval for Image and Video Databases VI*, volume 3312 of *Proceedings of SPIE*, pages 25–36, 1997.

48 Learning and Intelligent Image and Signal Analysis

Erkki Oja

The rapidly developing fields of neurocomputing and fuzzy logic, often combined under the term "computational intelligence" or "learning and intelligent systems", have proved their efficiency in a number of practical, hard real-world problems. Especially, signal processing, image analysis, pattern recognition, control, and fault diagnosis are the most central application fields of artificial neural networks and fuzzy systems in industry. This was one of the applications fields chosen in the national technology program "Adaptive and intelligent systems applications", launched by the Finnish Technology Development Centre (TEKES) in 1994.

The research activities in this field were collected under the multi-partner project "Learning and Intelligent Image and Signal Analysis" (LIISA), started in 1995 and ending in February 1998. The LIISA consortium was composed of university and VTT laboratories, selected for their proven ability in solving real world signal and image processing problems by neural and fuzzy techniques and in the formalization of the theoretical aspects in research; and industrial companies, selected for their experience in application development using the modern methodologies, and also for their knowledge of up-to-date industrial needs and capacities for offering challenging practical problems for piloting cases.

The LIISA project group was relatively large, consisting of *12 research laboratories* and *15 Finnish companies* in the fields of electronics, instrumentation, telecommunications, pulp and paper, steel manufacturing, and marketing. For this reason, and also because of the constraints set by the participating companies, it was divided in four *subprojects* (the main responsible university shown in brackets):

- Adaptive image analysis (Helsinki U. Tech.)
- Neuro-fuzzy systems (Tampere U. Tech.)
- Intelligent signal analysis and the required component technology (Oulu U.)
- Sensus - intelligent utilization of sensors by neural computing (Lappeenranta U. Tech.).

The major research partners were Helsinki University of Technology (Laboratory of Computer and Information Science), Tampere University of Technology, Lappeenranta University of Technology, and Oulu University. The responsible leader of the project was E. Oja from Helsinki University of Technology.

The project applied *neurocomputing and fuzzy logic in the estimation, preprocessing, detection, and classification of images and signals.* Typical applications were in industrial fault diagnosis and classification. In most cases, the measurement data was obtained from the industrial partners, and depending on the problem, this data was preprocessed and analyzed using neurocomputing and fuzzy logic tools. Because image and signal analysis is typically very application-oriented, the specific pre- and postprocessing algorithms had to be adapted to the problem. The generic

neural and fuzzy computing tools that were used throughout the project were the Self-Organizing Map (SOM) and the Learning Vector Quantization (LVQ) classifier, developed at Helsinki University of Technology by prof. T. Kohonen; the Principal Component Analysis (PCA) and Partial Least Squares (PLS) preprocessing methods and the related Averaged Learning Subspace Method (ALSM) classifier; the Multi-Layer Perceptron (MLP) and the Radial Basis Function (RBF) neural classifiers; and fuzzy logic including the trainable ANFIS neuro-fuzzy model. In the use and development of all these neural and fuzzy computing methods, the participating research groups have a long tradition. Thus also good quality software was already available in the laboratories, which helped in the fast start-up of the pilot projects.
Some pilot cases concentrated on hardware implementations. It is characteristic of the applications of real time image and signal processing that the computational requirements are very high. Some applications will be impossible without dedicated hardware, either based on VLSI or signal processors. Especially in the subproject on Neuro-fuzzy systems, there was a special task on developing a neural computer for industrial purposes, not only for the LIISA project but for the whole TEKES technology program.
As for the *main results*, working pilot applications were realized in all the subprojects to give guidelines to the companies for further development. The research phase terminated officially in February 1998, and the project has now entered a phase in which the companies are integrating the results into their own solution methods. Several product development projects are under way in the participating companies.
Research experience in the field was an important criterion for choosing the university partners, and one important output of the project were also the numerous research papers in journals and conferences. In addition to the research papers, several internal reports have been written within the project, and Ph. D. and M.Sc. theses have been completed on the results. In our research group, two doctoral degrees were attained within this project: Dr. Jorma Laaksonen in 1997 and Dr. Jukka Iivarinen in 1998. Detailed reports on the substance of this research are given as separate Sections in this report.
Starting in March 1998, a continuation of this consortium project was launched, the IMPRESS project (Intelligent Methods for Processing and Exploration of Signals and Systems). The total budget is about 11 million FIM and there are 17 companies involved. Our laboratory is involved in this research effort in several tasks, like on-line character recognition, fault analysis of a running paper web, and process monitoring and modelling using the SOM. There are reports on all of these activities in separate Sections.

49 Neural Methods for Analyzing Financial Information: How to Find the Enterprises with High Bankruptcy Risk?

Kimmo Kiviluoto, Erkki Oja and Jyrki Maaranen

Assessing the probability of bankruptcy of an enterprise is one of the key issues in a credit granting decision. Besides analyzing the strategy, personnel etc. of the firm, the financiers usually perform an analysis of the financial statements using some mathematical model. The standard approach has been to use a model based on Linear Discriminant Analysis, but a wide variety of other statistical techniques have also been proposed. Recently, models utilizing neural networks have been introduced and compared with the "traditional" techniques.

The importance of the problem has made it something of a benchmark test for different models. Usually, in these tests the problem has been reduced to a classification of companies into healthy and non-healthy ones. There are two characteristics common to many of the reported studies: they are based on fairly small data sets, and the proportion of the bankrupt firms is much higher in the data than in the total population, from which the sample is selected. This makes the results somewhat difficult to interpret – with small data sets, especially when the results are not cross-validated, the differences in classifier performance cannot be clearly distinguished from statistical noise, and with biased sample, one may also get an over-optimistic view of the classifier performance on the total population.

Another aspect is trying to analyze and understand the bankruptcy phenomenon: which factors contribute to an increased bankruptcy risk, or how does an increased risk of bankruptcy manifest itself?

The present study is conducted in co-operation with Finnvera Ltd., a service company that specializes in financing and development of small and medium-sized enterprises in Finland. The material consists of a certain segment of Finnvera Ltd.'s customer companies.

The study consists of two parts: qualitative analysis and classification. In the qualitative phase, the financial statements are analyzed with the Self-Organizing Map (SOM), which forms a "non-linear regression" from the input space into a plane. This makes it possible to visually examine the differences between firms that go bankrupt and those that do not (see figure 107) [1, 2, 3, 4, 5, 6]. New developments of the SOM are also discussed here, including three-dimensional SOM's, and a hierarchical model to analyze the year-to-year trajectories of an enterprise on the SOM [7, 8, 9, 10].

The classification of companies into healthy and non-healthy ones is done in two different ways: trying to minimize the total number of misclassifications, and using the Neyman-Pearson criterion, i.e. fixing the type I error (classifying a bankrupt company erroneously as a healthy company) to a suitable value, and with this constraint minimizing the type II error (classifying a healthy company erroneously as a bankrupt company). In practice, a classifier that is based on the Neyman-Pearson criterion would be the preferred one: type I error is much more costly than type II error, but because the proportion of non-bankrupt companies is higher, a classifier that minimizes the total number of misclassifications would pay more attention on minimizing the type II errors.

The classifiers used in the quantitative study are the following: Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), k-Nearest-Neighbour Classifier (kNN), Learning Vector Quantization (LVQ), Self-Organizing Map (SOM), and SOM-based Radial Basis Function Network (RBF-SOM). The modification of the LVQ algorithm to incorporate the Neyman-Pearson criterion is an original contribution of this study; the other methods are used in a fairly standard manner. The classification results are presented in tables 15 and 16 [1, 2, 6].

The results show that the Self-Organizing Map is a valuable tool in the qualitative analysis of the financial statement data. In classification, the LVQ and kNN classifiers performed best, when the aim was to minimize the total number of misclassifications. With the Neyman-Pearson criterion, the LDA classifier reached the level of the LVQ and kNN, with the SOM classifiers coming close to these.

In the future, the focus will shift from annually given financial statements to more frequent time series. A first step into this direction has been to analyze parallel sales time series with Independent Component Analysis (ICA) [11].

References

[1] K. Kiviluoto, "Analyzing financial statements with the self-organizing map," Master's thesis, Helsinki University of Technology, Espoo, Finland, May 1996.

[2] K. Kiviluoto and P. Bergius, "Analyzing financial statements with the self-organizing map," in *Proceedings of the workshop on self-organizing maps (WSOM'97)*, (Espoo, Finland), pp. 362–367, Neural Networks Research Centre, Helsinki University of Technology, June 1997.

[3] K. Kiviluoto and P. Bergius, *Visual Explorations in Finance using Self-Organising Maps*, ch. Maps for analysing failures of small and medium-sized enterprises. Springer, 1998.

[4] K. Kiviluoto and P. Bergius, "Exploring corporate bankruptcy with two-level self-organizing maps," in *Proceedings of the Fifth International Conference on Computational Finance*, (Boston, Massachusetts, USA), pp. 373–380, London Business School, Kluwer Academic Publishers, Dec. 1998.

[5] K. Kiviluoto and P. Bergius, "Two-level self-organizing maps for analysis of financial statements," in *Proceedings of the 1998 IEEE International Joint Conference on Neural Networks (IJCNN'98)*, vol. 1, (Piscataway, New Jersey, USA), pp. 189–192, IEEE Neural Networks Council, May 1998.

[6] K. Kiviluoto, "Predicting bankruptcies with the self-organizing map," *Neurocomputing*, vol. 21, pp. 191–201, 1998.

[7] K. Kiviluoto, "Topology preservation in self-organizing maps," in *Proceedings of the International Conference on Neural Networks (ICNN'96)*, vol. 1, (Piscataway, New Jersey, USA), pp. 294–299, IEEE Neural Networks Council, June 1996.

Table 15: Classification results using Neyman-Pearson criterion with different error I values (per cent), based on financial statements given 2 ... 0 years before bankruptcy

Classifier	error I target	total error	st.dev.	errorI	st.dev.	error II	st.dev.
LDA	0,20	19,0	(1,6)	21,0	(6,2)	18,8	(2,3)
	0,25	15,7	(1,0)	25,7	(5,4)	14,6	(1,5)
	0,30	14,1	(1,0)	29,5	(5,1)	12,5	(1,4)
LVQ	0,20	20,5	(2,0)	21,9	(4,1)	20,3	(2,4)
	0,25	15,9	(0,8)	25,7	(5,4)	14,9	(1,6)
	0,30	14,3	(1,0)	30,3	(4,5)	12,5	(1,5)
RBF-SOM	0,20	18,3	(1,2)	20,7	(5,6)	18,1	(1,7)
	0,25	15,8	(0,8)	26,4	(6,1)	14,7	(1,3)
	0,30	13,5	(1,0)	30,5	(6,4)	11,7	(1,6)
SOM	0,20	20,1	(1,9)	19,9	(6,3)	20,1	(2,6)
	0,25	16,6	(1,3)	25,4	(6,7)	15,7	(2,0)
	0,30	14,8	(0,4)	30,4	(6,8)	13,2	(0,9)

[8] K. Kiviluoto and E. Oja, "S-map: A network with a simple self-organization algorithm for generative topographic mappings," in *Advances in Neural Information Processing Systems* (M. I. Jordan, M. J. Kearns, and S. A. Solla, eds.), vol. 10, (Cambridge, Massachusetts, USA), pp. 549–555, MIT Press, 1998.

[9] K. Kiviluoto and E. Oja, "Softmax-network and S-Map – models for density-generating topographic mappings," in *Proceedings of the 1998 IEEE International Joint Conference on Neural Networks (IJCNN'98)*, vol. 3, (Piscataway, New Jersey, USA), pp. 2268–2272, IEEE Neural Networks Council, May 1998.

[10] K. Kiviluoto, "Comparing 2D and 3D self-organizing maps in financial data visualization," in *Methodologies for the Conception, Design and Application of Soft Computing – Proceedings of the 5th International Conference on Soft Computing and Information/Intelligent Systems (IIZUKA'98). Iizuka, Fukuoka, Japan.* (T. Yamakawa and G. Matsumoto, eds.), vol. 1, (Singapore), pp. 68–71, World Scientific, Oct. 1998.

[11] K. Kiviluoto and E. Oja, "Independent component analysis for parallel financial time series," in *ICONIP'98 Proceedings. Kitakyushu, Japan.* (S. Usui and T. Omori, eds.), vol. 2, (Tokyo, Japan.), pp. 895–898, APNNA, JNNS., Ohmsha, Oct. 1998.

Table 16: Classification results when minimizing the total number of misclassifications (per cent)

Classifier	total error	error I	error II
LVQ (e)	9,0	55,5	4,3
LVQ (p)	8,6	65,2	2,7
kNN (k=15)	8,5	75,2	1,5
LDA	10,5	47,1	6,6
QDA	11,1	55,9	6,5

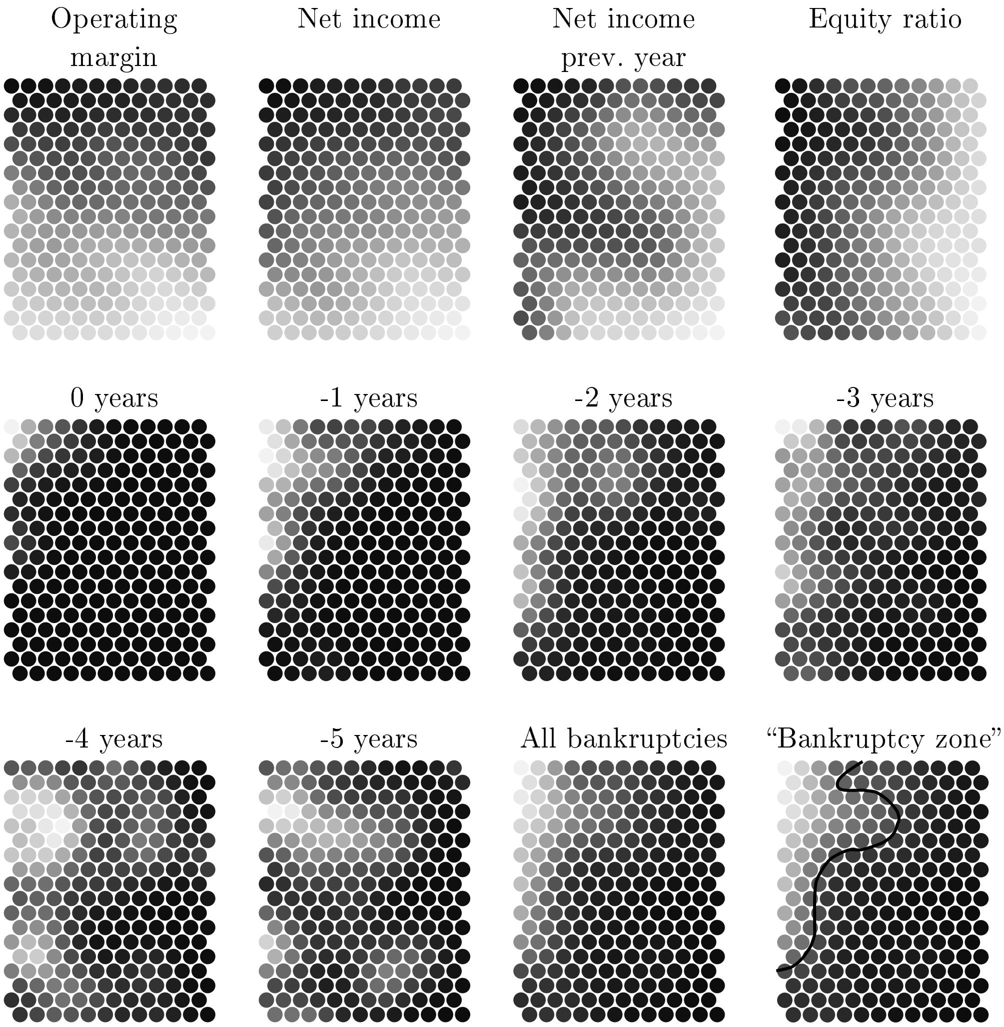


Figure 107: Financial indicators vs. bankruptcies, depicted 5 ... 0 years before the bankruptcy. On the upmost row, light color indicates good values of the financial indicators; on the two lower rows, light color indicates higher proportions of bankruptcy companies. In the lower right corner, a "bankruptcy zone" is drawn: more than one third of the companies that are projected on the left side of the line have gone bankrupt.

50 Methods for Decision Support in Diabetes Care

Mikko Mäkipää

In this project a data-driven approach for the decision support in diabetes care using artificial neural networks based on data collected by the patient during normal blood glucose self-monitoring was investigated. The focus was on insulin dependent diabetes mellitus (IDDM).

Diabetes is a major chronic disease that affects a growing number of people in the western industrialized countries. Recently, it has been shown that maintaining the blood glucose level as close to normal as possible has a dramatic effect in reducing the risk of long-term complications. As a major share of diabetes related health care costs is directed to the care of complications, improved blood glucose control would not only lead to improved quality of life for the patients and but potentially to substantial health care cost savings. Because of the complexity of the treatment required to maintain such benefical blood glucose levels, information technology offers a number of possibilities in supporting diabetes care.

As a part of the research a categorization of different approaches in decision support based on their applicability was proposed. The most potential areas of decision support where information technology could be applied were identified as first, the development of patient specific models for prediction of the blood glucose response for any given treatment and second, the analysis of collected self-monitoring data for therapy assessment. Neural network methods were futher developed on both these areas.

To build a model of an individual patient's glucose metabolism based on the measurement data for blood glucose level prediction a two-level approach was developed. Missing blood glucose values were estimated using gaussian mixture models and Expectation-Maximization (EM) algorithm. Multiply completed data sets were then used to train a committee of feed-forward neural networks. The prediction performance was evaluated using cross-validation. Evaluation results of the approach using a preliminary model and self-monitoring blood glucose data are promising.

As an application of retrospective data analysis, the Self-Organizing Map (SOM) was used for the clustering of daily therapy responses to find groups with similar blood glucose profiles, constituting a novel approach in diabetes data analysis. The SOM was found to be a particularly suitable method for forming the clustering as it is reliable, it can deal with missing values and produces ordered results. The method was again demonstrated on real patient data. In the tests, the formed groups showed clear and clinically interesting differences in all patients. Further, it seems that factors affecting the BG response, such as day-of-the-week and exercise, can be linked to the formed groups. The method could be applied to facilitate therapy analysis and discussion between patient and physician.

References

[1] Mäkipää, M. Neurocomputing methods in diabetes decision support. Master's Thesis, Helsinki University of Technology, Finland. December 1996.

51 Self-Organizing Map for Data Mining in MATLAB: the SOM Toolbox

Juha Vesanto, Esa Alhoniemi, Johan Himberg, Kimmo Kiviluoto, and Jukka Parviainen

The SOM Toolbox (`http://www.cis.hut.fi/projects/somtoolbox`) is a free function library for MATLAB 5 implementing the Self-Organizing Map (SOM) algorithm which is a neural network algorithm based on unsupervised learning [1]. Basically it performs a vector quantization and simultaneously organizes the quantized vectors on a regular low-dimensional grid. The SOM has proven to be a valuable tool in data mining because it is readily explainable, simple and easy to visualize. It has been successfully applied in various engineering applications in pattern recognition, image analysis, process monitoring and fault diagnosis [2, 3].

Thus far, the most useful implementation of the SOM and related tools has been the SOM_PAK (`http://www.cis.hut.fi/nnrc/nnrc-programs.html`). It is a public domain software package developed in the Neural Networks Research Centre of the Helsinki University of Technology, written in C language for UNIX and PC environments. However, the Mathwork Inc.'s MATLAB has been steadily gaining popularity as the "language of scientific computing". Moreover, MATLAB is much better-suited for fast prototyping and customizing than the C language used in SOM_PAK, as MATLAB employs a high-level programming language with strong support for graphics and visualization. All of these properties are extremely important in data mining. SOM Toolbox is an attempt to take full advantage of these strengths and provide an efficient, customizable and easy-to-use implementation of the SOM.

While closely related to SOM_PAK, SOM Toolbox is, however, a new set of programs. Both program packages have their relative strengths and weaknesses. The advantages of SOM_PAK are that it is written in ANSI C and thus runs in virtually any environment. It is an order of magnitude faster than SOM Toolbox in training. The advantages of SOM Toolbox are mainly in user friendliness and visualization capabilities. If desired, the SOM_PAK files can be accessed with the Toolbox: it is possible to first train the SOM with the SOM_PAK and then use the Toolbox for visualization.

SOM Toolbox utilizes MATLAB structures and the functions are constructed in a modular manner, which makes it convenient to tailor the code for each users' specific needs. The use of structs allows the Toolbox to keep track of many kinds of information that greatly facilitate the data mining process: labels associated with individual data vectors, variable names, data normalization information and training log.

The basic usage of the SOM Toolbox consists of three steps: SOM initialization, training and visualization. To make things easier to the user, the high-level functions require minimum number of parameters. For example, SOM size and training parameters are, unless specified, determined automatically based on the training data.

```
» sM=som_init(data); %initialization
» sM=som_train(sM,data); %training
» som_show(sM); %visualization, see Figure 108
```

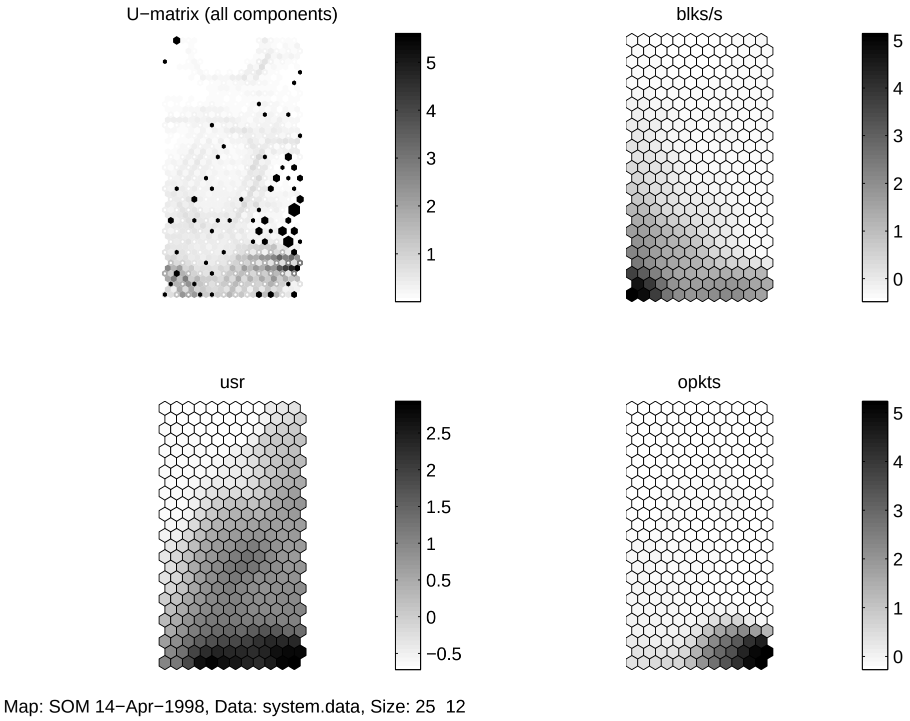


Figure 108: U-matrix and three components planes visualized by the SOM Toolbox. Hits from a small data set has been added on top of the U-matrix.

All this can also done through a graphical user interface. Around these three basic steps, SOM Toolbox has a large number of functions that can be used for preprocessing of the data and post-processing/analyzing the SOM.
We have found that the SOM Toolbox has greatly facilitated our research work. Implementation in MATLAB allows fast prototyping and powerful visualization. Building application specific tools on top of the Toolbox has proven to be easy.
Currently we are working on version 2 of the Toolbox. The major differences to the old version will be in visualization, which will utilize the newest research results in the field [4]. In addition, the package will include a larger set of supplementary algorithms and tools. Version 2 should be available during 1999.

References

[1] T. Kohonen. *Self-Organizing Maps.* Springer, Berlin, Heidelberg, 1995.

[2] T. Kohonen, E. Oja, O. Simula, A. Visa, and J. Kangas. Engineering applications of the self-organizing map. *Proceedings of the IEEE*, 84(10):27 pages, October 1996.

[3] O. Simula and J. Kangas. *Neural Networks for Chemical Engineers*, volume 6 of *Computer-Aided Chemical Engineering*, chapter 14, Process monitoring and visualization using self-organizing maps. Elsevier, Amsterdam, 1995.

[4] J. Vesanto. SOM-Based Data Visualization Methods. *Intelligent Data Analysis*, April 1999 (to appear).

Publications 1994 – 1998

Publications 1994 – 1998

1994

J. Iivarinen, K. Valkealahti, A. Visa, and O. Simula, Feature Selection with Self-Organizing Feature Map, Proc. of International Conference on Artificial Neural Networks, ICANN'94, Sorrento, Italy, May 26-29, 1994, pp. 334-337.

J. Iivarinen, T. Kohonen, J. Kangas, and S. Kaski, Visualizing the Clusters on the Self-Organizing Map, Proc. of Conference on Artificial Intelligence Research in Finland, STeP-94, Turku, Finland, August 29-31, 1994, pp. 122-126.

J. Joutsensalo, Nonlinear Data Compression and Representation by Combining Self-Organizing Map and Subspace Rule, Proc. of International Conference on Neural Networks, Orlando, Florida, June 26 - July 2, 1994, pp. 637-640.

J. Joutsensalo, High-Resolution Bearing Estimation by Fourier Methods, in Signal Processing VII: Theories and Applications, vol. II, pp. 637-640, M. Holt et al. (Eds.) (Proceedings of EUSIPCO-94, Edinburgh, Scotland, U.K., September 1994).

J. Joutsensalo, On the Fourier Methods for Estimating the Signal Subspace, Helsinki University of Technology, Laboratory of Computer and Information Science, Report A23, January 1994, 25 pages.

J. Kangas, Self-Organizing Maps in Error Tolerant Transmission of Vector Quantized Images, Helsinki University of Technology, Laboratory of Computer and Information Science, Report A21, 1993, 13 pages.

J. Kangas and T. Kohonen, Developments and Applications of the Self-Organizing Map and Related Algorithms, Proc. of SPRANN'94, IMACS International Symposium on Signal Processing, Robotics and Neural Networks, Lille, France, April 25-27, 1994, pp. 19-22.

J. Karhunen, Optimization Criteria and Nonlinear PCA Neural Networks, Proc. of International Conference on Neural Networks, Orlando, Florida, June 26 - July 2, 1994, pp. 1242-1246.

J. Karhunen, Stability of Oja's Subspace Rule, Neural Computation, vol. 6, pp. 739-747, 1994.

J. Karhunen and J. Joutsensalo, Representation and Separation of Signals Using Nonlinear PCA Type Learning, Neural Networks, vol. 7, pp. 113-127, 1994.

S. Kaski and T. Kohonen, Winner-Take-All Networks for Physiological Models of Competitive Learning, Neural Networks, vol. 7, nos. 6/7, pp. 973-984, 1994.

T. Kohonen, Physiological Model for the Self-Organizing Map, Proc. of World Congress on Neural Networks, 1994 International Neural Network Society Annual Meeting, San Diego, California, USA, June 4-9, 1994, pp. III-97 - III-102.

T. Kohonen, What Generalizations of the Self-Organizing Map Make Sense?, Proc. of International Conference on Artificial Neural Networks, ICANN'94, Sorrento, Italy, May 26-29, 1994, vol. 1, pp. 292-297.

M. Kurimo, Corrective Tuning by Applying LVQ for Continuous Density and Semi-

Continuous Markov Models, Proc. of International Symposium on Speech, Image Processing and Neural Networks, Hong Kong, April 1994, vol. 2, pp. 718-721.

M. Kurimo, Hybrid Training Method for Tied Mixture Density Hidden Markov Models Using Learning Vector Quantization and Viterbi Estimation, Proc. of IEEE Workshop on Neural Networks for Signal Processing, Ermioni, Greece, September 1994, pp. 362-371.

H. Kälviäinen, P. Hirvonen, L. Xu, and E. Oja, Comparisons of Probabilistic and Non-Probabilistic Hough Transforms, Proc. Third European Conference on Computer Vision, ECCV, Stockholm, Sweden, May 2-6, 1994, pp. 351-360.

I. Linnankoski, M. Laakso, R. Aulanko, and L. Leinonen, Recognition of Emotions in Macaque Vocalizations by Children and Adults, Language and Communication, vol. 14, pp. 183-192, 1994.

E. Oja, Neural Networks - Advantages and Applications, in E.S. Gelsema and L.N. Kanal (Eds.), Pattern Recognition in Practice IV (Amsterdam: Elsevier), pp. 359-365, 1994.

E. Oja, Beyond PCA: Statistical Expansions by Nonlinear Neural Networks, Proc. of International Conference on Artificial Neural Networks, ICANN'94, Sorrento, Italy, May 26-29, 1994, pp. 1049-1054.

E. Oja and J. Lampinen, Feature Extraction by Unsupervised Learning, in T. Ishiguro (Ed.), Cognitive Processing for Vision and Voice, Proc. Fourth NEC Research Symposium (Philadelphia: SIAM), pp. 63-76, 1994.

E. Oja and J. Lampinen, Unsupervised Learning for Feature Extraction, in J. M. Zurada, R.J. Marks II, and C.J. Robinson (Eds.), Computational Intelligence: Imitating Life (New York: IEEE Press), pp. 13-22, 1994.

K. Raivio and T. Kohonen, Detection of Nonlinearly Distorted and Two-Path Propagated Signals using SOM-Based Equalizers, Proc. of International Conference on Artificial Neural Networks, ICANN'94, Sorrento, Italy, May 26-29, 1994, vol. 2, pp. 1037-1040.

H. Rihkanen, L. Leinonen, T. Hiltunen, and J. Kangas, Spectral Pattern Recognition of Improved Voice Quality, Journal of Voice, vol. 8, pp. 320-326, 1994.

M. Vapola, O. Simula, T. Kohonen, and P. Meriläinen, Representation and Identification of Fault Conditions of an Anaesthesia System by Means of the Self-Organizing Map, Proc. of International Conference on Artificial Neural Networks, ICANN'94, Sorrento, Italy, May 26-29, 1994, pp. 350-353.

M. Vapola, O. Simula, T. Kohonen, and P. Meriläinen, Monitoring of an Anaesthesia System Using Self-Organizing Maps, Proc. of Conference on Artificial Intelligence Research in Finland, STeP-94, Turku, Finland, August 29-31, 1994, pp. 55-58.

A. Visa, Texture Segmentation Based on Neural Networks, Proc. of the 3rd International Conference on Fuzzy Logic, Neural Nets and Soft Computing, Iizuka, Japan, August 1-7, 1994, pp.145-149.

A. Visa, K. Valkealahti, J. Iivarinen, and O. Simula. Experiences from Operational Cloud Classifier Based on Self-Organising Map, SPIE vol. 2243 Applications of

Artificial Neural Networks V, pp. 484-495, Orlando, Florida, April 5-8, 1994.

1995

T. Honkela, V. Pulkki, and T. Kohonen, Contextual Relations of Words in Grimm Tales, Analyzed by Self-Organizing Map, Proc. International Conference on Artificial Neural Networks (ICANN'95), Paris, France, October 9-13, 1995, vol. 2, pp. 3-7.

J. Iivarinen, M. Peura, and A. Visa, Verification of a Multispectral Cloud Classifier, Proc. 9th Scandinavian Conference on Image Analysis, Uppsala, Sweden, June 6-9, 1995, vol. 1, pp. 591-599.

J. Iivarinen, K. Valkealahti, A. Visa, and O. Simula, Development of a Cloud Classifier, Report A25, Helsinki University of Technology, Faculty of Information Technology, Laboratory of Computer and Information Science, 1995, 49 p.

J. Joutsensalo and J. Karhunen, A Nonlinear Extension of the Generalized Hebbian Learning, Neural Processing Letters, vol. 2, no. 1, pp. 5-8, 1995.

J. Joutsensalo, A. Miettinen, and M. Zeindl, Nonlinear Dimension Reduction by Combining Competitive and Distributed Learning, Proc. 1995 International Conference on Artificial Neural Networks (ICANN'95), Paris, France, October 9-13, 1995, vol. 2, pp. 395-400.

J. Joutsensalo and A. Miettinen, Self-Organizing Operator Map for Nonlinear Dimension Reduction, Proc. International Conference on Neural Networks (ICNN'95), Perth, Australia, Nov. 27 - Dec. 1, 1995, vol. 1, pp. 111-114.

S.-L. Joutsiniemi, S. Kaski, and A. Larsen, Self-Organizing Map in Recognition of Topographic Patterns of EEG Spectra, IEEE Transactions on Biomedical Engineering, vol. 42, no. 11, pp. 1062-1068, 1995.

J. Kangas, Using Self-Organizing Map in Error Tolerant Transmission of Vector Quantized Images, Proc. World Congress on Neural Networks (WCNN'95), Washington, D.C., USA, July 17-21, 1995, pp. I-517-522.

J. Kangas, Increasing the Error Tolerance in Transmission of Vector Quantized Images by the Self-Organizing Maps, Proc. International Conference on Artificial Neural Networks (ICANN'95), Paris, France, October 9-12, 1995, vol. 1, pp. 287-291.

J. Kangas, Sample Weighting When Training Self-Organizing Maps for Image Compression, Proc. Neural Networks for Signal Processing V (NNSP'95), F. Girosi et al. (Eds.), IEEE, New York, 1995, pp. 343-350.

J. Kangas, Utilizing the Similarity Preserving Properties of Self-Organizing Maps in Vector Quantization of Images, Proc. IEEE International Conference on Neural Networks (ICNN'95), Perth, Australia, Nov. 27 - Dec. 1, 1995, vol. 4, pp. 2081-2084.

J. Karhunen and J. Joutsensalo, Generalizations of Principal Component Analysis, Optimization Problems, and Neural Networks, Neural Networks, vol. 8, no. 4, pp.

549-562, 1995.

J. Karhunen, E. Oja, L. Wang, R. Vigario, and J. Joutsensalo, A Class of Neural Networks for Independent Component Analysis, Report A28, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland, 1995, 29 p.

J. Karhunen, L. Wang, and J. Joutsensalo, Neural Estimation of Basis Vectors in Independent Component Analysis, Proc. International Conference on Artificial Neural Networks (ICANN'95), October 9-13, 1995, Paris, France, vol. 1, pp. 317-322.

J. Karhunen, L. Wang, and R. Vigario, Nonlinear PCA Type Approaches for Source Separation and Independent Component Analysis, Proc. International Conference. on Neural Networks (ICNN'95), Perth, Australia, Nov. 28 - Dec. 1, 1995, vol. 2, pp. 995-1000.

S. Kaski and T. Kohonen, Structures of Welfare and Poverty in the World Discovered by the Self-Organizing Map, Report A24, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland, 1995, 17 p.

K. Kiviluoto, Topology Preservation in Self-Organizing Maps, Report A29, Helsinki University of Technology, Faculty of Information Technology, Laboratory of Computer and Information Science, Espoo, Finland, 1995, 8 p.

T. Kohonen, Self-Organizing Maps, Springer-Verlag, Series in Information Sciences, vol. 30, Heidelberg, 1995.

T. Kohonen, Emergence of Invariant-Feature Detectors in Self-Organization, in Computational Intelligence, A Dynamic System Perspective, M. Palaniswami, Y. Attikiouzel, R.J. Marks II, D. Fogel, and T. Fukuda (Eds.), IEEE Press, 1995, Chapter 2, pp. 17-31.

T. Kohonen, Aivoalueiden ja muistin teoria, in Tutkimuksen etulinjassa, J. Rydman (Ed.), WSOY, Porvoo, 1995, pp. 251-262 (in Finnish).

T. Kohonen, The Adaptive-Subspace SOM (ASSOM) and its Use for the Implementation of Invariant Feature Detection, Proc. International Conference on Artificial Neural Networks (ICANN'95), Paris, France, October 9-13, 1995, vol. 1, pp. 3-10.

T. Kohonen, Learning Vector Quantization, in The Handbook of Brain Theory and Neural Networks, M.A. Arbib (Ed.), The MIT Press, Cambridge, Massachusetts, 1995, pp. 537-540.

T. Kohonen, Method for Controlling an Electronic Musical Device by Utilizing Search Arguments and Rules to Generate Digital Code Sequences, US Patent 5,418,323, May 23, 1995.

T. Kohonen (inventor), Oy Nokia Ab (assignee), Adaptive Detection Method and Detector for Quantized Signals, US Patent 5,428,644, June 27, 1995.

H. Kälviäinen, P. Hirvonen, and E. Oja, Houghtool – A Software Package for Hough Transform Calculation, Proc. 9th Scandinavian Conference on Image Analysis, Uppsala, Sweden, June 6-9, 1995, pp. 841-848.

H. Kälviäinen, P. Hirvonen, L. Xu, and E. Oja, Probabilistic and Non-Probabilistic Hough Transforms: Overview and Comparisons, Image and Vision Computing, vol. 13, pp. 239-252, 1995.

J. Lampinen and E. Oja, Distortion Tolerant Pattern Recognition Based on Self-Organizing Feature Extraction, IEEE Transactions on Neural Networks, vol. 6, pp. 539-547, 1995.

E. Oja, Principal Component Analysis, in The Handbook of Brain Theory and Neural Networks, M. Arbib (Ed.), MIT Press, Cambridge, 1995, pp. 753-756.

E. Oja, Unsupervised Neural Learning, in Neural Networks for Chemical Engineers, A. Bulsari (Ed.), Elsevier, Amsterdam, 1995, pp. 21-32.

E. Oja, The Nonlinear PCA Learning Rule and Signal Separation - Mathematical Analysis, Helsinki University of Technology, Laboratory of Computer and Information Science, Report A26, 1995, 24 p.

E. Oja, PCA, ICA, and Nonlinear Hebbian Learning, Proc. International. Conference on Artificial Neural Networks (ICANN'95), Paris, France, October 9-13, 1995, vol. 1, pp. 89-94.

E. Oja and J. Karhunen, Signal Separation by Nonlinear Hebbian Learning, in Computational Intelligence, A Dynamic System Perspective, M. Palaniswami, Y. Attikiouzel, R.J. Marks II, D. Fogel, and T. Fukuda (Eds.), IEEE Press, 1995, Chapter 6, pp. 83-97.

E. Oja and O. Taipale, Applications of Learning and Intelligent Systems - the Finnish Technology Programme, Proc. International Conference on Artificial Neural Networks (ICANN'95), Industrial Session 1 (Neural Network Clubs & Funding Programme), Paris, France, Oct. 9-13, 1995.

E. Oja and K. Valkealahti, Compressing Higher-Order Co-Occurrences for Texture Analysis Using the Self-Organizing Map, Proc. International Conference. on Neural Networks (ICNN'95), Perth, Australia, Nov. 28 - Dec. 1, 1995, vol. 2, pp. 1160-1164.

P. Pajunen, J. Joutsensalo, J. Karhunen, and K. Saarinen, Estimation of Equispaced Sinusoids Using Maximum Likelihood Method, Proc. 1995 Finnish Signal Processing Symposium (FINSIG'95), Espoo, Finland, June 1995, pp. 128-132.

P. Pajunen, J. Joutsensalo, J. Karhunen, and K. Saarinen, Maximum Likelihood Estimation of Equispaced Sinusoids in Rotating Machine Fault Detection, Proc. 1995 International Conference on Signal Processing Applications and Technology (ICSPAT'95), Boston, USA, October 1995, pp. 1164-1168.

T. Pessi, J. Kangas, and O. Simula, Patient Grouping Using Self-Organizing Map, Proc. International Conference on Artificial Neural Networks (ICANN'95), Industrial Session 5 (Medicine), Paris, France, October 9-12, 1995.

V. Pulkki, Data Averaging inside Categories with the Self-Organizing Map, Report A27, Helsinki University of Technology, Faculty of Information Technology, Laboratory of Computer and Information Science, Espoo, Finland, 1995.

K. Raivio, J. Henriksson, and O. Simula, Neural Detection of QAM Modulation in the Presence of Interference, Proc. IEEE International Conference on Neural Networks (ICNN'95), Perth, Australia, Nov. 27 - Dec. 1, 1995, vol. 4, pp. 1566-1569.

K. Raivio, J. Henriksson, and O. Simula, Interference Cancellation for PAM Modulation using Neural Networks, Proc. of the Finnish Signal Processing Symposium, Espoo, Finland, June 2, 1995, pp. 50-54.

O. Simula and J. Kangas, Process Monitoring and Visualization Using Self-Organizing Maps, in Neural Networks for Chemical Engineers, A. Bulsari (Ed.), Elsevier Science B.V., Amsterdam, 1995, pp. 371-384.

H. Tang and O. Simula, The Optimal Utilization of Multi-Service SCP, Proc. IFIP-TC6 Working Conference on Intelligent Systems, Copenhagen, Denmark, August 28-31, 1995, pp. 157-167.

H. Tang and O. Simula, Neural Adaptation for Optimal Traffic Shaping in Telephone Systems, Proc. 1995 IEEE International Conference on Neural Networks (ICNN'95), Perth, Australia, Nov. 27 - Dec. 1, 1995, vol. 4, pp. 1561-1565.

K. Torkkola and T. Kohonen, Speech Recognition: A Hybrid Approach, in The Handbook of Brain Theory and Neural Networks, M.A. Arbib (Ed.), The MIT Press, Cambridge, Massachusetts, 1995, pp. 907-910.

K. Valkealahti and A. Visa, Simulated Annealing in Feature Weighting for Classification with Learning Vector Quantization, Proc. 9th Scandinavian Conference on Image Analysis, Uppsala, Sweden, June 6-9, 1995, vol. 2, pp. 965-971.

R. Vigario and E. Oja, Signal Separation and Feature Extraction by Nonlinear PCA Network, Proc. 9th Scandinavian Conference on Image Analysis, Uppsala, Sweden, June 6-9, 1995, pp. 811-818.

A. Visa, J. Iivarinen, K. Valkealahti, and O. Simula, Neural Network Based Cloud Classifier, Proc. International Conference on Artificial Neural Networks (ICANN'95), Industrial Session 14 (Remote Sensing), Paris, France, October 9-13, 1995.

J. Vuori and T. Kohonen, Fast DSP Implementation of High-Dimensional Vector Classifier, Proc. 1995 IEEE International Conference on Neural Networks (IC-NN'95), Perth, Australia, Nov. 27 - Dec. 1, 1995, vol. 4, pp. 2019-2022.

L. Wang, J. Karhunen, and E. Oja, A Bigradient Optimization Approach for Robust PCA, MCA, and Source Separation, Proc. International Conference on Neural Networks (ICNN'95), Perth, Australia, Nov. 28 - Dec. 1, 1995, vol. 3, pp. 1684-1689.

L. Wang, J. Karhunen, E. Oja, and R. Vigario, Blind Separation of Sources Using Nonlinear PCA Type Learning Algorithms, Proc. International Conference on Neural Networks and Signal Processing (ICNNSP'95), Nanjing, P.R. China, Dec. 10-13, 1995, pp. 847-850.

1996

E. Alhoniemi, O. Simula, and J. Vesanto, Monitoring and Modeling of Complex Processes Using the Self-Organizing Map, Proc. of the International Conference on Neural Information Processing (ICONIP'96), Hong Kong, September 24-27, 1996, vol. 2, pp. 1169-1174.

M.-L. Haapanen, L. Liu, T. Hiltunen, L. Leinonen, and J. Karhunen, Cul-de-sac Hypernasality Test with Pattern Recognition of LPC Indices, Folia Phoniatrica et Logopaedica, vol. 48, pp. 35-43, 1996.

J. Hollmén and O. Simula, Prediction Models and Sensitivity Analysis of Industrial Process Parameters by Using the Self-Organizing Map, Proc. IEEE Nordic Signal Processing Symposium (NORSIG'96), Espoo, Finland, September 24-27, 1996, pp. 79-82.

L. Holmström, P. Koistinen, J. Laaksonen, and E. Oja, Neural Network and Statistical Perspectives of Classification, Proc. 13th International Conference on Pattern Recognition, Vienna, Austria, Aug. 25-30, 1996, vol. IV, pp. 286-290.

L. Holmström, P. Koistinen, J. Laaksonen, and E. Oja, Comparison of Neural and Statistical Classifiers - Theory and Practice, University of Helsinki, Rolf Nevanlinna Institute, Research Reports A13, 1996, 37 p.

T. Honkela, S. Kaski, K. Lagus, and T. Kohonen, Self-Organizing Maps of Document Collections, Polysteekki, Bulletin of the Helsinki University of Technology, Suppl., English edition, pp. 20-22, 1996.

T. Honkela, S. Kaski, K. Lagus, and T. Kohonen, Exploration of Full-Text Databases with Self-Organizing Maps, Proc. IEEE International Conference on Neural Networks (ICNN'96), Washington, D.C., USA, June 2-6, 1996, vol. 1, pp. 56-61.

T. Honkela, S. Kaski, K. Lagus, and T. Kohonen, Newsgroup Exploration with WEBSOM Method and Browsing Interface, Report A32, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland, 1996, 13 p.

J. Hurri, A. Hyvärinen, J. Karhunen, E. Oja, Image Feature Extraction Using Independent Component Analysis, Proc. of the 1996 IEEE Nordic Signal Processing Symposium (NORSIG'96), Espoo, Finland, September 24-27, 1996, pp. 475-478.

A. Hyvärinen, Purely Local Neural Principal Component and Independent Component Learning, Proc. International Conference on Artificial Neural Networks (ICANN'96), Bochum, Germany, July 16-19, 1996, Lecture Notes in Computer Science, vol. 1112, pp. 139-144, Springer, Berlin, 1996.

A. Hyvärinen, Simple One-Unit Neural Algorithms for Blind Source Separation and Blind Deconvolution, Proc. International Conference on Neural Information Processing, Hong Kong, September 24-27, 1996, pp. 1201-1206.

A. Hyvärinen, Finding Cluster Directions by Non-linear Hebbian Learning, Proc. International Conference on Neural Information Processing, Hong Kong, September 24-27, 1996, pp. 97-102.

A. Hyvärinen, A Family of Fixed-Point Algorithms for Independent Component Analysis, Report A40, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland, 1996, 16 p.

A. Hyvärinen and E. Oja, A Neuron That Learns to Separate One Signal from a Mixture of Independent Sources, Proc. International Conference on Neural Networks (ICNN'96), Washington DC, USA, June 3-6, 1996, pp. 62-67.

A. Hyvärinen and E. Oja, A Fast Fixed-Point Algorithm for Independent Component Analysis, Helsinki University of Technology, Laboratory of Computer and Information Science, Report A35, 1996, 11 p.

A. Hyvärinen and E. Oja, Independent Component Analysis by General Non-linear Hebbian-like Learning Rules, Helsinki University of Technology, Laboratory of Computer and Information Science, Report A41, 1996, 11 p.

A. Hyvärinen and E. Oja, Simple Neuron Models for Independent Component Analysis, Helsinki University of Technology, Laboratory of Computer and Information Science, Report A37, 1996, 25 p.

J. Iivarinen, J. Rauhamaa, and A. Visa, An Adaptive Approach to Segmentation of Surface Defects, Technical Report A34, Helsinki University of Technology, Laboratory of Computer and Information Science, March 1996, 15 p.

J. Iivarinen, J. Rauhamaa, and A. Visa, Unsupervised Segmentation of Surface Defects, Proc. 13th International Conference on Pattern Recognition, Wien, Austria, August 25-30, 1996, vol. IV, pp. 356-360.

J. Iivarinen and A. Visa, Shape Recognition of Irregular Objects, in Intelligent Robots and Computer Vision XV: Algorithms, Techniques, Active Vision, and Materials Handling, Proc. SPIE 2904, pp. 25-32, Boston, Massachusetts, November 19-21, 1996.

J. Joutsensalo, A Subspace Method for Model Order Estimation in CDMA, Proc. IEEE Fourth International Symposium on Spread Spectrum Techniques and Applications (ISSSTA'96), Mainz, Germany, September 22-25, 1996, vol. 2, pp. 688-692.

J. Joutsensalo and A. Alastalo, Linear Model Order Estimation by Signal Subspaces, Proc. IEEE Nordic Signal Processing Symposium (NORSIG'96), Espoo, Finland, September 24-27, 1996, pp. 319-322.

J. Joutsensalo, J. Lilleberg, A. Hottinen, and J. Karhunen, A Hierarchic Maximum Likelihood Method for Delay Estimation in CDMA, Proc. of the IEEE Vehicular Technology Conference, Atlanta, Georgia, USA, April 28 - May 1, 1996, pp. 188-192.

J. Joutsensalo, J. Lilleberg, A. Hottinen, and J. Karhunen, Hierarchic Method for CDMA Synchronization, Proc. of the 1996 IEEE Nordic Signal Processing Symposium (NORSIG'96), Espoo, Finland, September 24-27, 1996, pp. 17-20.

J. Kangas and S. Kaski, Compression of Vector Quantization Code Sequences Based on Code Frequencies and Spatial Redundancies, Proc. IEEE International Conference on Image Processing (ICIP'96), Lausanne, Switzerland, September 16-19, 1996, vol. III, pp. 463-466, IEEE Service Center, Piscataway, NJ, 1996.

J. Kangas and T. Kohonen, Developments and Applications of the Self-Organizing

Map and Related Algorithms, Mathematics and Computers in Simulation, vol. 41, no. 5-6, pp. 3-12, 1996.

J. Karhunen, Neural Approaches to Independent Component Analysis and Source Separation, Proc. of the 4th European Symposium on Artificial Neural Networks (ESANN'96), Bruges, Belgium, April 24-26, 1996, pp. 249-266.

J. Karhunen and P. Pajunen, Hierarchic Nonlinear PCA Algorithms for Neural Blind Source Separation, Proc. of the 1996 IEEE Nordic Signal Processing Symposium (NORSIG'96), Espoo, Finland, September 24-27, 1996, pp. 71-74.

S. Kaski, Computationally Efficient Approximation of a Probabilistic Model for Document Representation in the WEBSOM Full-Text Analysis Method, Report A38, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland, 1996, 10 p.

S. Kaski, T. Honkela, K. Lagus, and T. Kohonen, Creating an Order in Digital Libraries with Self-Organizing Maps, Proc. World Congress on Neural Networks (WCNN'96), San Diego, Calif., USA, September 15-18, 1996, pp. 814-817.

S. Kaski and K. Lagus, Comparing Self-Organizing Maps, in C. von der Malsburg, W. von Seelen, J. C. Vorbrüggen, and B. Sendhoff (Eds.), Proc. International Conference on Artificial Neural Networks (ICANN'96), Bochum, Germany, July 16-19, Lecture Notes in Computer Science, vol. 1112, pp. 809-814, Springer, Berlin, 1996.

S. Kaski and T. Kohonen, Exploratory Data Analysis by the Self-Organizing Map: Structures of Welfare and Poverty in the World, in Neural Networks in Financial Engineering, A.-P.N. Refenes, Y. Abu-Mostafa, J. Moody, and A. Weigend (Eds.), World Scientific, Singapore, 1996, pp. 498-507.

K. Kiviluoto, Topology Preservation in Self-Organizing Maps, Proc. International Conference on Neural Networks (ICNN'96), Washington, D.C., USA, June 3-6, 1996, IEEE Neural Networks Council, Piscataway, New Jersey, USA, pp. 294-299.

T. Kohonen, Emergence of Invariant-Feature Detectors in the Adaptive-Subspace Self-Organizing Map, Biological Cybernetics, vol. 75, pp. 281-291, 1996.

T. Kohonen, Emergence of Invariant-Feature Detectors in the Adaptive-Subspace Self-Organizing Maps, Proc. 1996 IEEE Nordic Signal Processing Symposium (NORSIG'96), September 24-27, 1996, Espoo, Finland, pp. 65-70.

T. Kohonen, New Developments and Applications of Self-Organizing Maps, Proc. 1996 International Workshop on Neural Networks for Identification, Control, Robotics, and Signal/Image Processing (NICROSP'96), Venice, Italy, August 21-23, 1996, IEEE Computer Society Press, Los Alamitos, Calif., pp. 164-172.

T. Kohonen, The Speedy SOM, Report A33, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland, 1996, 7 p.

T. Kohonen, Advances in the Development and Application of Self-Organizing Maps, Proc. 5th Turkish Symposium on Artificial Intelligence and Neural Networks (TAINN'96), E. Alpaydin et al. (Eds.), Istanbul, Turkey, June 27-28, 1996, pp. 3-12.

T. Kohonen, The Self-Organizing Map, a Possible Model of Brain Maps, Medical & Biological Engineering & Computing, vol. 34, Supplement 1, Part 1, pp. 5-8, 1996 (Papers of the 10th Nordic-Baltic Conference on Biomedical Engineering, Tampere, Finland, June 9-13, 1996).

T. Kohonen, Avaako neurolaskenta oven virtuaalimaailmaan?, Futura, no. 1, pp. 7-11, 1996 (in Finnish).

T. Kohonen, Self-Organizing Maps of Symbol Strings, Report A42, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland, 1996, 35 p.

T. Kohonen, J. Hynninen, J. Kangas, and J. Laaksonen, SOM_PAK: The Self-Organizing Map Program Package, Report A31, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland, 1996, 25 p.

T. Kohonen, J. Hynninen, J. Kangas, J. Laaksonen, and K. Torkkola, LVQ_PAK: The Learning Vector Quantization Program Package, Report A30, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland, 1996, 26 p.

T. Kohonen, S. Kaski, K. Lagus, and T. Honkela, Very Large Two-Level SOM for the Browsing of Newsgroups, Proc. of International Conference on Artificial Neural Networks (ICANN'96), Bochum, Germany, July 16-19, C. von der Malsburg, W. von Seelen, J.C. Vorbrüggen, and B. Sendhoff (Eds.), Lecture Notes in Computer Science, vol. 1112, Springer, pp. 269- 274.

T. Kohonen, E. Oja, O. Simula, A. Visa, and J. Kangas, Engineering Applications of the Self-Organizing Map, Proc. IEEE, vol. 84, no. 10, pp. 1358-1384, 1996.

M. Kurimo, Segmental LVQ3 Training for Phoneme-Wise Tied Mixture Density HMMs, in Signal Processing VIII, Proc. 8th European Signal Processing Conference, Trieste, Italy, September 10-13, 1996, Ramponi, G., Sicuranza, G.L., Carrato, S., and Marsi, S. (Eds.), Edizioni Lint Trieste, Trieste, Italy, 1996, pp. 1599-1602.

M. Kurimo and P. Somervuo, Using the Self-Organizing Map to Speed up the Probability Density Estimation for Speech Recognition with Mixture Density HMMs, Proc. of 4th International Conference on Spoken Language Processing, Philadelphia, Pennsylvania, October 3-6, 1996, pp. 358-361.

H. Kälviäinen, P. Hirvonen, and E. Oja, Houghtool – A Software Package for the Use of the Hough Transform, Pattern Recognition Letters, vol. 17, pp. 889-897, 1996.

J. Laaksonen and E. Oja, Classification with Learning k-Nearest Neighbors, Proc. International Conference on Neural Networks (ICNN'96), Washington DC, USA, June 3-6, 1996, vol. 3, pp. 1480-1483.

J. Laaksonen and E. Oja, Subspace Dimension Selection and Averaged Learning Subspace Method in Handwritten Digit Classification, Proc. International Conference on Artificial Neural Networks (ICANN'96), Bochum, Germany, July 16-19, 1996, pp. 227-232.

K. Lagus, T. Honkela, S. Kaski, and T. Kohonen, WEBSOM - A Status Report, Proc. STeP-96 – Genes, Nets and Symbols, Finnish Artificial Intelligence Conference, Vaasa, Finland, August 20-23, 1996, pp. 73-78.

K. Lagus, T. Honkela, S. Kaski, and T. Kohonen, Self-Organizing Maps of Document Collections: A New Approach to Interactive Exploration, Proc. Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA, August 2-4, 1996, E. Simoudis, J. Han, U. Fayada (Eds.), AAAI Press, Menlo Park, Calif., 1996, pp. 238-242.

K. Lagus, S. Kaski, T. Honkela, and T. Kohonen, Browsing Digital Libraries with the Aid of Self-Organizing Maps, Proc. Fifth International World Wide Web Conference WWW5, Paris, France, May 6-10, 1996, Poster Proceedings, pp. 71-79.

H. Lappalainen, Soft Multiple Winners for Sparse Feature Extraction, Proc. International Conference on Neural Networks (ICNN'96), Washington, D.C., USA, June 3-6, 1996, pp. 207-210.

L. Leinonen, K. Valkealahti, H. Rihkanen, Visual Imaging of Voice Quality with the Self-Organizing Map, Suomen logopedis-foniatrinen aikakauslehti, vol. 16, pp. 89-96, 1996.

E. Oja and A. Hyvärinen, Blind Signal Separation by Neural Networks, Proc. 1996 International Conference on Neural Information Processing (ICONIP'96), Hong Kong, September 24-27, 1996, pp. 7-14.

E. Oja and L. Wang, Robust Fitting by Nonlinear Neural Units, Neural Networks, vol. 9, pp. 435-444, 1996.

E. Oja and L. Wang, Neural Fitting: Robustness by Anti-Hebbian Learning, Neurocomputing, vol. 12, pp. 155-170, 1996.

E. Oja and K. Valkealahti, Co-Occurrence Map: Quantizing Multidimensional Texture Histograms, Pattern Recognition Letters, vol. 17, pp. 723-730, 1996.

P. Pajunen, Nonlinear Independent Component Analysis by Self-Organizing Maps, in Artificial Neural Networks, Proc. International Conference on Artificial Neural Networks (ICANN'96), Bochum, Germany, July 16-19, 1996, C. von der Malsburg et al. (Eds.), pp. 815-819, Springer, 1996.

P. Pajunen, An Algorithm for Binary Blind Source Separation, Technical Report A36, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland, 1996, 10p.

P. Pajunen, A. Hyvärinen, and J. Karhunen, Nonlinear Blind Source Separation by Self-Organizing Maps, Proc. of the 1996 International Conference on Neural Information Processing (ICONIP'96), Hong Kong, September 24-27, 1996, pp. 1207-1210.

O. Simula, E. Alhoniemi, J. Hollmen, and J. Vesanto, Monitoring and Modeling of Complex Processes Using Hierarchical Self-Organizing Maps, Proc. of 1996 IEEE International Symposium on Circuits and Systems (ISCAS-96), Atlanta, Georgia, USA, May 12-15, 1996, Supplement to vol. 4, pp. 73-76.

J. Sinkkonen, S. Kaski, M. Huotilainen, R.J. Ilmoniemi, R. Näätänen, and K. Kaila, Optimal Resource Allocation for Novelty Detection in a Human Auditory Memory, Neuroreport, vol. 7, pp. 2479-2482, 1996.

A. Stocker, O. Sipilä, A. Visa, O. Salonen, and T. Katila, Stability Study of Some Neural Networks Applied to Tissue Characterization of Brain Magnetic Resonance Images, Proc. 13th International Conference on Pattern Recognition, Vienna, Austria, August 25-29, 1996, vol. IV, pp. 472-477.

H. Tang and O. Simula, The Adaptive Resource Assignment and Optimal Utilization of Multi-Service SCP, 4th International Conference on Intelligence in Networks, Bordeaux, France, November 25-28, 1996, pp. 235-240.

H. Tang and O. Simula, The Optimal Resource Management of a Network Server, Second International Symposium on Operations Research and its Applications, Guilin, China, December 11-14, 1996, pp. 397-406.

H. Tang and O. Simula, The Optimal Utilization of Multiservice SCP, in Intelligent Networks and New Technologies, J. Norgaard and V. B. Iversen (Eds.), Chapman & Hall, London, 1996, pp. 175-188.

H. Tang, O. Simula, and K. Raatikainen, Age-Boosting Page Replacement Scheme, Report A39, Helsinki University of Technology, Laboratory of Computer and Information Science, 1996, 12 p.

K. Valkealahti and E. Oja, Optimal Texture Feature Selection for the Co-Occurrence Map, Proc. International Conference on Artificial Neural Networks (ICANN'96), Bochum, Germany, July 16 - 19, 1996, pp. 245-250.

R. Vigario, A. Hyvärinen, and E. Oja, ICA Fixed-Point Algorithm in Extraction of Artifacts from EEG, Proc. 1996 IEEE Nordic Signal Processing Symposium (NORSIG'96), Espoo, Finland, Sep. 24-27, 1996, pp. 383-386.

L. Wang, and J. Karhunen, A Unified Neural Bigradient Algorithm for Robust PCA and MCA, International Journal of Neural Systems, vol. 7, no. 1, March 1996, pp. 53-67.

1997

Alhoniemi, E., Himberg, J., Kiviluoto, K., Parviainen, J. & Vesanto, J. SOM Toolbox for Matlab 5, version 1.0β, November 7, 1997.
http://www.cis.hut.fi/projects/somtoolbox/.

Henriksson, J. & Raivio, K. Finnish patent No. 98177. Method and Circuit Arrangement for Processing a Signal Containing Interference, 1997, 27 p.

Himberg, J. & Simula, O. Analyzing an Automatic Call Distribution System by the Self-Organizing Map. Proc. of 1997 Finnish Signal Processing Symposium (FINSIG'97), Pori, Finland, May 22, 1997, pp. 153-157.

Holmström, L., Koistinen, P., Laaksonen, J. & Oja, E. Neural and Statistical Classifiers - Taxonomy and Two Case Studies. IEEE Transactions on Neural Networks, 1997, Vol. 8, No. 1, pp. 5-17.

Honkela, T. Comparisons of Self-Organized Word Category Map. Proc. of Workshop on Self-Organizing Maps (WSOM'97), Espoo, Finland, June 4-6, 1997, pp. 298-303.

Honkela, T. Self-Organizing Maps of Words for Natural Language Processing Applications. Proc. of International ICSC Symposium on Soft Computing, Nimes, France, September 17-19, 1997, pp. 401-407.

Honkela, T., Kaski, S., Lagus, K. & Kohonen, T. WEBSOM - Self- Organizing Maps of Document Collections. Proc. of Workshop on Self- Organizing Maps (WSOM'97), Espoo, Finland, June 4-6, 1997, pp. 310-315.

Honkela, T., Lehtola, A., Kalliomäki, S., Suitiala, R., Hudson, R., Karkaletsis, V. & Vouros, G. A Recommended Globalization Method. In: Hall, P.A.V. & Hudson, R. (eds.), Software Without Frontiers. Chichester 1997, John Wiley and Sons, pp. 33-50.

Hurri, J., Hyvärinen, A. & Oja, E. Wavelets and Natural Image Statistics. Proc. of 10th Scandinavian Conference on Image Analysis, Lappeenranta, Finland, June 9-12, 1997, pp. 13-18.

Hyvärinen, A. A Family of Fixed-Point Algorithms for Independent Component Analysis. Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'97), Munich, Germany, April 21-24, 1997, pp. 3917-3920.

Hyvärinen, A. One-Unit Contrast Functions for Independent Component Analysis: A Statistical Analysis. Proc. of IEEE Workshop on Neural Networks for Signal Processing '97, Amelia Island, Florida, USA, Sept. 24- 26, 1997, pp. 388-397.

Hyvärinen, A. & Oja, E. A Fast Fixed-Point Algorithm for Independent Component Analysis. Neural Computation, 1997, Vol. 9, No. 7, pp. 1483- 1492.

Hyvärinen, A. & Oja, E. One-Unit Learning Rules for Independent Component Analysis. Proc. of Neural Information Processing Systems (NIPS'96), Denver, Colorado, Dec. 2-5, 1996. Cambridge, MA 1997, The MIT Press, pp. 480-486.

Hyvärinen, A. Independent Component Analysis by Minimization of Mutual Information. Helsinki University of Technology, Laboratory of Computer and Information Science, Report A46, Espoo, Finland, 1997, 35 p.

Hyvärinen, A. New Approximations of Differential Entropy for Independent Component Analysis and Projection Pursuit. Helsinki University of Technology, Laboratory of Computer and Information Science, Report A47, Espoo, Finland, 1997, 10 p.

Iivarinen, J., Peura, M., Särelä, J. & Visa, A. Comparison of Combined Shape Descriptors for Irregular Objects. Proc. of 8th British Machine Vision Conference, University of Essex, UK, September 8-11, 1997, pp. 430-439.

Iivarinen, J., Rauhamaa, J. & Visa, A. An Adaptive Two-Stage Approach to Classification of Surface Defects. Proc. of 10th Scandinavian Conference on Image Analysis, Lappeenranta, Finland, June 9-11, 1997, Vol. 1, pp. 317- 322.

Joutsensalo, J. Algorithms for Delay Estimation and Tracking in CDMA. Proc. of IEEE International Conference on Communications (ICC'97), Montreal, Quebec, Canada, June 8-12, 1997, pp. 366-370.

Joutsensalo, J. Semi-Blind Source Parameter Separation. Proc. of International

Conference on Artificial Neural Networks (ICANN'97), Lausanne, Switzerland, October 8-10, 1997, pp. 577-582.

Joutsensalo, J., Lilleberg, J., Hottinen, A. & Karhunen, J. Subspace Algorithms for Synchronization and Tracking in CDMA. Proc. of First IEEE Signal Processing Workshop on Signal Processing Advances in Wireless Communications (SPAWC'97), Paris, France, April 16-18, 1997, pp. 369- 372.

Kallioniemi, I., Saarinen, J. & Oja, E. Extraction of the Opto-Geometrical Properties of Gratings from Scattering Data by Means of the Neural Network. Proc. of Diffractive Optics, Savonlinna, Finland, July 7-9, 1997, pp. 202-203.

Karhunen, J., Cichocki, A., Kasprzak, W. & Pajunen, P. On Neural Blind Separation with Noise Suppression and Redundancy Reduction. International Journal of Neural Systems, 1997, Vol. 8, No. 2, pp. 219-237.

Karhunen, J., Hyvärinen, A., Vigário, R., Hurri, J. & Oja, E. Applications of Neural Blind Separation to Signal and Image Processing. Proc. of 1997 International Conference on Acoustics, Speech, and Signal Processing (ICASSP-97), Munich, Germany, April 21-24, 1997, pp. 131-134.

Karhunen, J., Oja, E., Wang, L., Vigário, R. & Joutsensalo, J. A Class of Neural Networks for Independent Component Analysis. IEEE Transactions on Neural Networks, 1997, Vol. 8, No. 3, pp. 486-504.

Karhunen, J. & Pajunen, P. Blind Source Separation Using Least-Squares Type Adaptive Algorithms. Proc. of IEEE 1997 International Conference on Acoustics, Speech, and Signal Processing (ICASSP-97), Munich, Germany, April 21-24, 1997, pp. 3361-3364.

Karhunen, J. & Pajunen, P. Blind Source Separation and Tracking Using Nonlinear PCA Criterion: A Least-Squares Approach. Proc. of IEEE 1997 International Conference on Neural Networks (ICNN'97), Houston, Texas, June 9-12, 1997, pp. 2147-2152.

Karjalainen, M., Boda, P., Somervuo, P. & Altosaar, T. Applications for the Hearing-Impaired: Evaluation of Finnish Phoneme Recognition Methods. Proc. of 5th European Conference on Speech Communication and Technology, Rhodes, Greece, September 22-25, 1997, pp. 1811-1814.

Karkaletsis, V., Spyroupoulos, C., Vouros, G., Honkela, T., Lagus, K. & Lehtola, A. Message Generation. In: Hall, P.A.V. & Hudson, R., Software Without Frontiers. Chichester 1997, John Wiley and Sons, pp. 203-218.

Kaski, S. Computationally Efficient Approximation of a Probabilistic Model for Document Representation in the WEBSOM Full-Text Analysis Method. Neural Processing Letters, 1997, Vol. 5, pp. 139-151.

Kiviluoto, K. & Bergius, P. Analyzing Financial Statements with the Self- Organizing Map. Proc. of Workshop on Self-Organizing Maps (WSOM'97), Espoo, Finland, June 4-6, 1997, pp. 362-367.

Kohonen, T., Kaski, S. & Venna, J. Automatic Coloring of Data According to Its Cluster Structure. Helsinki University of Technology, Laboratory of Computer and

Information Science, Report A 45, Espoo, Finland, 1997, 12 p.

Kohonen, T., Kaski, S., Lappalainen, H. & Salojärvi, J. The Adaptive- Subspace Self-Organizing Map (ASSOM). Proc. of Workshop on Self- Organizing Maps (WSOM'97), Espoo, Finland, June 4-6, 1997, pp. 191-196.

Kohonen, T. Exploration of Large Document Collections by Self-Organizing Maps. Proc. of 6th Scandinavian Conference on Artificial Intelligence (SCAI'97), Helsinki, Finland, August 18-20, 1997. Amsterdam, Netherlands 1997, IOS Press, pp. 5-7.

Kohonen, T. & Somervuo, P. Self-Organizing Maps of Symbol Strings with Application to Speech Recognition. Proc. of Workshop on Self-Organizing Maps (WSOM'97), Espoo, Finland, June 4-6, 1997, pp. 2-7.

Kohonen, T. Exploration of Very Large Databases by Self-Organizing Maps. Proc. of International Conference on Neural Networks (ICNN'97), Houston, Texas, USA, June 9-12, 1997, pp. PL1-PL3.

Kohonen, T. Emergence of Optimal Invariant-Feature Detectors in a New Neural Network Architecture. Proc. of Fuzzy-Neuro-Systeme '97 - Computational Intelligence (FNS'97), Soest, Germany, March 12-14, 1997, p. 44.

Kohonen, T., Kaski, S. & Lappalainen, H. Self-Organized Formation of Various Invariant-Feature Filters in the Adaptive-Subspace SOM. Neural Computation, 1997, Vol. 9, pp. 1321-1344.

Kokkotos, S., Spyroupoulos, C., Honkela, T., Käpylä, T., Lagus, K. & Hall, P. Languages and Character Sets. In: Hall, P.A.V. & Hudson, R. (eds.), Software Without Frontiers. Chichester 1997, John Wiley and Sons, pp. 135- 158.

Kurimo, M. Comparison Results for Segmental Training Algorithms for Mixture Density HMMs. Proc. of 5th European Conference on Speech Technology and Communication, EUROSPEECH'97, Rhodes, Greece, September 22-25, 1997, pp. 87-90.

Kurimo, M. SOM Based Density Function Approximation for Mixture Density HMMs. Proc. of Workshop on Self-Organizing Maps (WSOM'97), Espoo, Finland, June 4-6, 1997, pp. 8-13.

Kurimo, M. Training Mixture Density HMMs with SOM and LVQ. Helsinki University of Technology, Laboratory of Computer and Information Science Report A43, Espoo, Finland, 1997, 26 p.

Laaksonen, J. Local Subspace Classifier and Local Subspace SOM. Proc. of Workshop on Self-Organizing Maps (WSOM'97), Espoo, Finland, June 4-6, 1997, pp. 32-37.

Laaksonen, J. Local Subspace Classifier. Proc. of International Conference on Artificial Neural Networks (ICANN'97), Lausanne, Switzerland, October 8-10, 1997, pp. 637-642.

Laaksonen, J. & Oja, E. Density Function Interpretation of Subspace Classification Methods. Proc. of 10th Scandinavian Conference on Image Analysis, June 9-12, 1997, Lappeenranta, Finland, pp. 487 - 492.

Laaksonen, J. & Oja, E. Error-Corrective Feature Extraction in Handwritten Digit Recognition. Proc. of Engineering Applications of Neural Networks (EANN97), June 16-18, 1997, Stockholm, Sweden, pp. 37-40.

Lagus, K. Map of WSOM'97 Abstracts - Alternative Index. Proc. of Workshop on Self-Organizing Maps (WSOM'97), Espoo, Finland, June 4-6, 1997, pp. 368-372.

Lagus, K., Suitiala, R. & Honkela, T. Culture, Conventions and Local Practices. In: Hall, P.A.V. & Hudson, R. (eds.), Software Without Frontiers. Chichester 1997, John Wiley and Sons, pp. 159-166.

Lampinen, J., Laaksonen, J. & Oja, E. Neural Network Systems, Techniques and Applications in Pattern Recognition. Helsinki University of Technology, Laboratory of Computational Engineering, Report B1, Espoo, Finland, 1997, 61 p.

Lehtola, A., Kalliomäki, S., Honkela, T. & Lillqvist, T. An Application Framework for Internationalization. In: Hall, P.A.V. & Hudson, R., Software Without Frontiers. Chichester 1997, John Wiley and Sons, pp. 83-110.

Leinonen, L., Hiltunen, T., Linnankoski, I. & Laakso, M.-L. Expression of Emotional-Motivational Connotations with a One-Word Utterance. Journal of the Acoustical Society of America, 1997, Vol. 102, No. 3, pp. 1853-1863.

Leinonen, L., Hiltunen, T., Laakso, M.-L., Rihkanen, H. & Poppius, H. Categorization of Voice Disorders with Six Perceptual Dimensions. Folia Phoniatrica Logopedica, 1997, Vol. 49, pp. 9-20.

Leinonen, L. & Poppius, H. Voice Reactions to Histamine Inhalation in Asthma. Allergy, 1997, Vol. 52, pp. 27-31.

Luukkanen, P. & Joutsensalo, J. Comparison of MUSIC and Matched Filter Delay Estimators in DS-CDMA. Proc. of 1997 International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC'97), Helsinki, Finland, September 1-4, 1997, pp. 830-834.

McHugh, N., Honkela, T. & Hudson, R. Quality Assurance. In: Hall, P.A.V. & Hudson, R., Software Without Frontiers. Chichester 1997, John Wiley and Sons, pp. 219-228.

Mäkipää, M., Heinonen, P. & Oja, E. Using the Self-Organizing Map in Supporting Diabetes Therapy. Proc. of Workshop on Self-Organizing Maps (WSOM'97), Espoo, Finland, June 4-6, 1997, pp. 51-56.

Oja, E. The Nonlinear PCA Learning Rule in Independent Component Analysis. Neurocomputing, 1997, Vol. 17, pp. 25-45.

Oja, E., Karhunen, J. & Hyvärinen, A. From Neural Principal Components to Neural Independent Components. Proc. of International Conference on Artificial Neural Networks (ICANN'97), Lausanne, Switzerland, October 8- 10, 1997, pp. 519-528.

Oja, E., Karhunen, J., Hyvärinen, A., Vigário, R. & Hurri, J. Neural Independent Component Analysis - Approaches and Applications. In: Kasabov, N. & Amari, S. (eds.), Brain-like Computing and Intelligent Information Systems. Berlin 1997, Springer.

Oja, E. & Valkealahti, K. Local Independent Component Analysis by the Self-Organizing Map. Proc. of International Conference on Artificial Neural Networks (ICANN'97), Lausanne, Switzerland, October 8-10, 1997, pp. 553- 558.

Padgett, M.L., Werbos, P.J. & Kohonen, T. Strategies and Tactics for the Application of Neural Networks to Industrial Electronics. In: Irwin, J.D. (ed.), Industrial Electronics Handbook. Boca Raton, Florida 1997, CRC Press, pp. 835-852.

Pajunen, P. Blind Separation of Binary Sources with less Sensors than Sources. Proc. of 1997 International Conference on Neural Networks (ICNN97), Houston, Texas, USA, June 9-12, 1997, pp. 1994-1997.

Pajunen, P. A Competitive Learning Algorithm for Separating Binary Sources. Proc. of European Symposium on Artificial Neural Networks (ESANN97), Brugge, Belgium, April 16-17, 1997, pp. 255-260.

Pajunen, P. & Karhunen, J. Self-Organizing Maps for Independent Component Analysis. Proc. of Workshop on Self-Organizing Maps (WSOM'97), Espoo, Finland, June 4-6, 1997, pp. 96-99.

Pajunen, P. & Karhunen, J. A Maximum Likelihood Approach to Nonlinear Blind Source Separation. Proc. of International Conference on Artificial Neural Networks (ICANN'97), Lausanne, Switzerland, October 8-10, 1997, pp. 541-546.

Pajunen, P. & Karhunen, J. Least-Squares Methods for Blind Source Separation Based on Nonlinear PCA, International Journal of Neural Systems, 1997, Vol. 8, No. 5-6, pp. 601-612.

Peura, M. A Statistical Classification Method for Hierarchical Irregular Objects. Proc. of 9th International Conference on Image Analysis and Processing (ICIAP'97), Firenze, Italy, September 17-19, 1997. Image Analysis and Processing, Lecture Notes in Computer Science, A. del Bimbo (ed.), 1997, Springer Verlag, Vol. 1, pp. 604-611.

Peura, M. & Visa, A. Computational Intelligence in Cloud Classification. Electronic Imaging Newsletter, SPIE and IS&T, 1997, Vol. 7, No. 1, pp. 4-8.

Peura, M. & Iivarinen, J. Efficiency of Simple Shape Descriptors. Proc. of 3rd International Workshop on Visual Form (IWVF3), Capri, Italy, May 28- 30, 1997. Singapore 1997, World Scientific, pp. 443-451.

Raivio, K., Hämäläinen, A., Henriksson, J. & Simula, O. Performance of Two Neural Receiver Structures in the Presence of Co-Channel Interference. Proc. of International Conference on Neural Networks (ICNN'97), Houston, Texas, June 9-12, 1997, pp. 2080-2084.

Raivio, K., Henriksson, J. & Simula, O. Neural Detection of QAM Signal with Strongly Nonlinear Receiver. Proc. of Workshop on Self-Organizing Maps (WSOM'97), Espoo, Finland, June 4-6, 1997, pp. 20-25.

Raivio, K., Henriksson, J. & Simula, O. Neural Receiver Structures Based on Self-Organizing Maps in Nonlinear Multipath Channels. Proc. of 1997 International Workshop on Applications of Neural Networks to Telecommunications (IWANNT), Melbourne, Australia, June 9-11, 1997, pp. 241-247.

Simula, O., Alhoniemi, E., Hollmén, J. & Vesanto, J. Analysis of Complex Systems

Using the Self-Organizing Map. Proc. of 4th International Conference on Neural Information Processing (ICONIP'97), Dunedin, New Zealand, November 24-28, 1997, pp. 1313-1317.

Sinkkonen, J. & Kaski, S. Topographic Components Model with Adaptive Latencies. Proc. of Third International Conference on Functional Mapping of the Human Brain, Copenhagen, Denmark, May 19-23, 1997.

Sinkkonen, J. & Kaski, S. Topographic Components Model with Adaptive Latencies. Neuroimage (Supplement), 1997, Vol. 5, No. 4, p. S445.

Tang, H., Simula, O. & Raatikainen, K. Age-Boosting Page Replacement Scheme. Proc. of International Conference on Telecomputers (ICT'97), Melbourne, Australia, April 2-4, 1997, Vol. 3, pp. 1115-1120.

Tang, H. & Simula, O. The Effective Resource Demands of the Applications and Their Managements in a Computer. Proc. of 3rd Asia-Pasific Conference on Communications (APPC'97), Sydney, Australia, December 7- 10, 1997, pp. 267-261.

Valkealahti, K. Texture Classification with Single- and Double-Resolution Co-Occurrence Maps. Proc. of International Conference on Engineering Applications of Neural Networks, Stockholm, Sweden, June 16-18, 1997, pp. 63-66.

Valkealahti, K. & Oja, E. Reduced Multidimensional Texture Histograms. Proc. of 10th Scandinavian Conference on Image Analysis, June 9-12, 1997, Lappeenranta, Finland, pp. 923-930.

Vesanto, J. Using the SOM and Local Models in Time-Series Prediction. Proc. of Workshop on Self-Organizing Maps (WSOM'97), Espoo, Finland, June 4-6, 1997, pp. 209-214.

Vigário, R. Extraction of Ocular Artefacts from EEG Using Independent Component Analysis. Electroencephalography and clinical Neuro- physiology, 1997, Vol. 103, No. 3, pp. 395-404.

Visa, A. & Iivarinen, J. Evolution and Evaluation of a Trainable Cloud Classifier. IEEE Transactions on Geoscience and Remote Sensing, 1997, Vol. 35, No. 5, pp. 1307-1315.

1998

Flanagan, J. A. Sufficient Conditions for Self-Organisation in the One-Dimensional SOM with a Reduced Width Neighbourhood. Neurocomputing, 1998, Vol. 21, pp. 51-60.

Flanagan, J. A. The Self-Organising Map, Robustness, Self-Organising Criticality and Power Laws. Proc. of European Symposium on Artificial Neural Networks (ESANN'98), Bruges, Belgium, April 22-24, 1998, pp. 209-214.

Himberg, J. Enhancing the SOM-based Data Visualization by Linking Different Data Projections. Proc. of 1st International Symposium IDEAL'98, Intelligent Data Engineering and Learning - Perspectives on Financial Engineering and Data Mining, Hong Kong, October 14-16, 1998, Springer, pp. 427-434.

Honkela, T., Lagus, K. & Kaski, S. Self-Organizing Maps of Large Document Collections. In: Deboeck, G. & Kohonen, T. (eds.), Visual Explorations in Finance with Self-Organizing Maps. London 1998, Springer, pp. 168-178.

Honkela, T., Kaski, S., Kohonen, T. & Lagus, K. Self-Organizing Maps of Very Large Document Collections: Justification for the WEBSOM Method. In: Balderjahn, I., Mathar, R. & Schader, M. (eds.), Classification, Data Analysis, and Data Highways. Berlin 1998, Springer, pp. 245-252.

Hurri, J., Gävert, H., Särelä, J. & Hyvärinen, A. FastICA Package for Matlab. Computer program. 1998. http://www.cis.hut.fi/projects/ica/fastica/.

Hyvärinen, A. Denoising of Sensory Data by Maximum Likelihood Estimation of Sparse Components. Proc. of International Conference on Artificial Neural Networks (ICANN'98), Skövde, Sweden, September 2-4, 1998, pp. 141-146.

Hyvärinen, A. Independent Component Analysis for Time-Dependent Stochastic Processes. Proc. of International Conference on Artificial Neural Networks (ICANN'98), Skövde, Sweden, September 2-4, 1998, pp. 135-140.

Hyvärinen, A. Independent Component Analysis in the Presence of Gaussian Noise by Maximizing Joint Likelihood. Neurocomputing, 1998, Vol. 22, pp. 49-67.

Hyvärinen, A. Independent Component Analysis in the Presence of Noise: A Maximum Likelihood Approach. Proc. of Workshop on Independence and Artificial Neural Networks (I&ANN'98), Tenerife, Spain, February 9-10, 1998, pp. 32-38.

Hyvärinen, A. New Approximations of Differential Entropy for Independent Component Analysis and Projection Pursuit. In: Jordan, M.I., Kearns, M.J. & Solla, S.A. (eds.), Advances in Neural Information Processing 10 (NIPS'97, Denver, Colorado, December 2-4, 1997). Cambridge, MA 1998, MIT Press, pp. 273-279.

Hyvärinen, A. Noisy Independent Component Analysis, Maximum Likelihood Estimation, and Competitive Learning. Proc. of IEEE International Joint Conference on Neural Networks (IJCNN'98), Anchorage, Alaska, May 4-9, 1998, pp. 2282-2287.

Hyvärinen, A. Sparse Code Shrinkage: Denoising of Nongaussian Data by Maximum Likelihood Estimation. Report A51, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland, 1998.

Hyvärinen, A. Sparse Regression: Utilizing the Higher-Order Structure of Data for Prediction. Proc. of International Conference on Artificial Neural Networks (ICANN'98), Skövde, Sweden, September 2-4, 1998, pp. 541-546.

Hyvärinen, A. Statistical Estimation in Conceptual Spaces. Proc. of International Conference on Artificial Neural Networks (ICANN'98), Skövde, Sweden, September 2-4, 1998, pp. 1071-1076.

Hyvärinen, A. & Honkela, T. Maladaptive Emotion-Based Behaviors in Autonomous Agents. Proc. of STeP'98, The 8th Finnish Artificial Intelligence Conference, Jyväskylä, Finland, September 7-9, 1998, pp. 218-226.

Hyvärinen, A., Hoyer, P. & Oja, E. Sparse Code Shrinkage for Image Denoising. Proc. of International Joint Conference on Neural Networks (IJCNN'98), Anchorage, Alaska, May 4-9, 1998, pp. 859-864.

Hyvärinen, A. & Oja, E. Independent Component Analysis by General Nonlinear Hebbian-Like Learning Rules. Signal Processing, 1998, Vol. 64, nro 3, pp. 301-313.

Hyvärinen, A., Oja, E., Hoyer, P. & Hurri, J. Image Feature Extraction by Sparse Coding and Independent Component Analysis. Proc. of 14th International Conference on Pattern Recognition (ICPR'98), Brisbane, Australia, August 17-20, 1998, pp. 1268-1273.

Hyvärinen, A. & Pajunen, P. On Existence and Uniqueness of Solutions in Nonlinear Independent Component Analysis. Proc. IEEE International Joint Conference on Neural Networks (IJCNN'98), Anchorage, Alaska, May 4-9, 1998, pp. 1350-1354.

Iivarinen, J. & Visa, A. Unsupervised Image Segmentation with the Self-Organizing Map and Statistical Methods. In: Casasent, D.P. (ed.), Intelligent Robots and Computer Vision XVII: Algorithms, Techniques, and Active Vision, Proc. SPIE 3522, pp. 516-526, 1998.

Iivarinen, J. & Rauhamaa, J. Surface Inspection of Web Materials Using the Self-Organizing Map. In: Casasent, D.P. (ed.), Intelligent Robots and Computer Vision XVII: Algorithms, Techniques, and Active Vision, Proc. SPIE 3522, pp. 96-103, 1998.

Iivarinen, J. & Visa, A. An Adaptive Texture and Shape Based Defect Classification. Proc. of 14th International Conference on Pattern Recognition, Brisbane, Australia, August 16-20, 1998. Vol. I, pp. 117-122.

Karhunen, J., Pajunen, P. & Oja, E. The Nonlinear PCA Criterion in Blind Source Separation: Relations with Other Approaches. Neurocomputing, 1998, Vol. 22, pp. 5-20.

Karhunen, J. & Pajunen, P. New Results on Nonlinear PCA Criterion in Blind Source Separation. Proc. of Workshop on Independence and Artificial Neural Networks (I&ANN'98), Tenerife, Spain, February 9-10, 1998, pp. 5-11.

Giannakopoulos, X., Karhunen, J. & Oja, E. An Experimental Comparison of Neural ICA Algorithms. Proc. of International Conference on Artificial Neural Networks (ICANN'98), Skövde, Sweden, September 2-4, 1998. Berlin 1998, Springer, pp. 651-656.

Karhunen, J., Pajunen, P. & Hyvärinen, A. Extending Neural ICA and Blind Source Separation for Nonlinear Data Models. Proc. of Workshop on Machines That Learn, Snowbird, Utah, USA, April 7-10, 1998, 2 p.

Cichocki, A., Karhunen, J., Kasprzak, W. & Vigário, R. Neural Networks for Blind Separation with Unknown Number of Sources. Helsinki University of Technology, Publications in Computer and Information Science, Report A54, Espoo, Finland, 1998, 46 p.

Karhunen, J. Book Review. Principal Component Neural Networks - Theory and Applications by K.I. Diamantaras and S.Y. Kung (New York, John Wiley 1996). Pattern Analysis and Applications, 1998, Vol. 1, pp. 74-75.

Kaski, S., Honkela, T., Lagus, K. & Kohonen, T. WEBSOM - Self-Organizing Maps of Document Collections. Neurocomputing, 1998, Vol. 21, pp. 101-117.

Kangas, J. & Kaski, S. 3043 Works that Have Been Based on the Self-Organizing Map (SOM) Method Developed by Kohonen. Report A49, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland, 1998, 193 p.

Kaski, S. & Kohonen, T. Tips for Processing and Color-Coding of Self-Organizing Maps. In: Deboeck, G. & Kohonen, T. (eds.), Visual Explorations in Finance with Self-Organizing Maps. London 1998, Springer, pp. 195-202.

Kaski, S., Nikkilä, J. & Kohonen, T. Methods for Interpreting a Self-Organized Map in Data Analysis. Proc. of 6th European Symposium on Artificial Neural Networks (ESANN'98), Bruges, Belgium, April 22-24, 1998. Brussels, Belgium 1998, D-Facto, pp. 185-190.

Kaski, S. Dimensionality Reduction by Random Mapping: Fast Similarity Computation for Clustering. Proc. of International Joint Conference on Neural Networks (IJCNN'98), Anchorage, Alaska, May 4-9, 1998. Piscataway, NJ 1998, IEEE Service Center, vol. 1, pp. 413-418.

Kaski, S., Lagus, K., Honkela, T. & Kohonen, T. Statistical Aspects of the WEBSOM System in Organizing Document Collections. Proc. of 29th Symposium on the Interface, Houston, Texas, May 14-17, 1997. Fairfax Station, VA 1998, Interface Foundation of North America, vol. 29, pp. 281-290.

Kaski, S., Kangas, J. & Kohonen, T. Bibliography of Self-Organizing Map (SOM) Papers: 1981-1997. Neural Computing Surveys, 1998, Vol. 1, nro 3&4, pp. 1-176.

Kiviluoto, K. & Oja, E. S-Map: A Network with a Simple Self-Organization Algorithm for Generative Topographic Mappings. In: Jordan, M.I., Kearns, M.J. & Solla, S.A. (eds.), Advances in Neural Information Processing 10 (NIPS'97, Denver, Colorado, USA, December 2-4, 1997). Cambridge, MA 1998, The MIT Press, pp. 549-555.

Kiviluoto, K. & Oja, E. Softmax-Network and S-Map - Models for Density-Generating Topographic Mappings. Proc. of International Joint Conference on Neural Networks (IJCNN'98), Anchorage, Alaska, May 4-9, 1998, pp. 2268-2272.

Kiviluoto, K. & Oja, E. Independent Component Analysis for Parallel Financial Time Series. Proc. of International Conference on Neural Information Processing (ICONIP'98), Kitakyushu, Japan, October 21-23, 1998, pp. 895-898.

Kiviluoto, K. Comparing 2D and 3D Self-Organizing Maps in Financial Data Visualization. Proc. of International Conference on Soft Computing and Information / Intelligent Systems (IIZUKA'98), Iizuka, Japan, October 16-20, 1998, pp. 68-71.

Kiviluoto, K. Predicting Bankruptcies with the Self-Organizing Map. Neurocomputing, 1998, Vol. 21, pp. 191-201.

Kiviluoto, K. & Bergius, P. Two-Level Self-Organizing Maps for Analysis of Financial Statements. Proc. of International Joint Conference on Neural Networks (IJCNN'98), Anchorage, Alaska, May 4-9, 1998, pp. 189-192.

Kiviluoto, K. & Bergius, P. Exploring Corporate Bankruptcy with Two-Level Self-Organizing Maps. Decision Technologies for Computational Management Science, Proc. of Fifth International Conference on Computational Finance, London, December 15-17, 1997. Boston 1998, Kluwer Academic Publishers, pp. 373-380.

Kiviluoto, K. & Bergius, P. Maps for Analyzing Failures of Small and Medium-sized Enterprises. In: Deboeck, G. & Kohonen, T. (eds.), Visual Expolorations in Finance with Self-Organizing Maps. London 1998, Springer, pp. 59-71.

Kohonen, T. The Self-Organizing Map. Neurocomputing, 1998. Vol. 21, pp. 1-6.

Kohonen, T. & Somervuo, P. Self-Organizing Maps of Symbol Strings. Neurocomputing, 1998. Vol. 21, pp. 19-30.

Kohonen, T. Computation of VQ and SOM Point Densities Using the Calculus of Variations. Report A52, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland, 1998.

Kohonen, T. The Self-Organizing Map Algorithms and their Applications in Science and Technology. Proc. of 6th European Congress on Intelligent Techniques & Soft Computing (EUFIT'98), Aachen, Germany, September 7-10, 1998. Aachen, Germany 1998, Elite Foundation, pp. 193-194.

Kohonen, T. Self-Organization of Very Large Document Collections: State of the Art. Proc. of 8th International Conference on Artificial Neural Networks (ICANN'98), Skövde, Sweden, September 2-4, 1998, pp. 65-74.

Kohonen, T. The SOM Methodology. In: Deboeck, G. & Kohonen, T. (eds.), Visual Explorations in Finance with Self-Organizing Maps. London 1998, Springer-Verlag, pp. 159-167.

Deboeck, G. & Kohonen, T. (eds.) Visual Explorations in Finance with Self-Organizing Maps. London 1998, Springer-Verlag. 258 p.

Kohonen, T. & Oja, E. Visual Feature Analysis by the Self-Organising Maps. Neural Computing & Applications, 1998. Vol. 7, pp. 273-286.

Kurimo, M. Self-Organization in Mixture Densities of HMM Based Speech Recognition. Proc. of European Symposium on Artificial Neural Networks (ESANN'98), Bruges, Belgium, April 22-24, 1998, pp. 237-242.

Kurimo, M. Improving Vocabulary Independent HMM Decoding Results by Using the Dynamically Expanding Context. Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'98), Seattle, WA, USA, May 12-15, 1998, pp. 833-836.

Lampinen, J., Laaksonen, J. & Oja, E. Pattern Recognition. In: Leondes, C.T. (ed.), Image Processing and Pattern Recognition. New York 1998, Academic Press, pp. 1-59.

Laaksonen, J. & Oja, E. Learning Subspace Classification and Error-Corrective Feature Extraction. International Journal of Pattern Recognition and Artificial Intelligence, 1998. Vol. 12, nro 4, pp. 423-436.

Laaksonen, J., Hurri, J., Oja, E. & Kangas, J. Experiments with a Self-Supervised Adaptive Classification Strategy in On-Line Recognition of Handwritten Latin Characters. Proc. of 6th Workshop on Frontiers of Handwriting Recognition, Taejon, Korea, August 12-14, 1998, pp. 475-484.

Laaksonen, J., Hurri, J., Oja, E. & Kangas, J. Comparison of Adaptive Strategies for On-Line Character Recognition. Proc. of International Conference on Neural Networks (ICANN'98), Skövde, Sweden, September 2-4, 1998, pp. 245-250.

Lagus, K. Generalizability of the WEBSOM Method to Document Collections of Various Types. Proc. of 6th European Congress on Intelligent Techniques & Soft Computing (EUFIT'98), Aachen, Germany, September 7-10, 1998, pp. 210-214.

Lappalainen, H. Using an MDL-Based Cost Function with Neural Networks. Proc. of International Joint Conference on Neural Networks (IJCNN'98), Anchorage, Alaska, May 4-9, 1998, pp. 2384-2389.

Mäkipää, M., Ebeling, P., Heinonen, P. & Oja, E. Estimation of Glycolated Hemoglobin from Blood Glucose Measurement Data. Proc. of Computers in Diabetes, Barcelona, Spain, September 6-7, 1998.

Oja, E., Karhunen, J., Hyvärinen, A., Vigário, R. & Hurri, J. Neural Independent Component Analysis - Approaches and Applications. In: Kasabov, N. & Amari, S. (eds.), Brain-like Computing and Intelligent Information Systems. Singapore 1998, Springer, pp. 167-188.

Oja, E. & Hyvärinen, A. Independent Components in Image Processing. Proc. of SSAB Symposium on Image Analysis, Uppsala, Sweden, March 16-17, 1998, pp. 1-4.

Oja, E. & Karhunen, J. Information Theoretic and Nonlinear PCA Approaches to Independent Component Analysis. Workshop on Machines That Learn, Snowbird, Utah, April 7-10, 1998, 2 p.

Kallioniemi, I., Saarinen, J. & Oja, E. Optical Scatterometry of Subwavelength Diffraction Gratings: Neural Networks Approach. Applied Optics, 1998. Vol. 37, nro 25, pp. 5830-5835.

Oja, E. From Neural Learning to Independent Components. Neurocomputing, 1998. Vol. 22, pp. 187-200.

Oja, E. ICA Learning Rules: Stationary, Stability, and Sigmoids. Proc. of International Workshop on Independence and Artificial Neural Networks (I&ANN'98), La Laguna, Spain, February 9-10, 1998, pp. 87-103.

Oja, E. The Nonlinear PCA Approach to ICA. Proc. of International Conference on Neural Information Processing (ICONIP'98), Kitakyushu, Japan, October 21-23, 1998, pp. 725-728.

Oja, E. Signal Decomposition by FastICA. Proc. of International Conference on Neural Information Processing (ICONIP'98), Kitakyushu, Japan, October 21-23, 1998, pp. 594-602.

Joutsensalo, J. & Pajunen, P. Blind Symbol Learning Algorithm for CDMA Systems. Proc. of IEEE International Joint Conference on Neural Networks (IJCNN'98), Anchorage, Alaska, May 4-9, 1998, pp. 152-155.

Joutsensalo, J. & Pajunen, P. Symbol Separation in Oversaturated CDMA System. Proc. of Fifth International Symposium on Spread Spectrum Techniques and Applications (ISSSTA'98), Sun City, South Africa, September 2-4, 1998, pp. 498-501.

Koivunen, V., Pajunen, P., Karhunen, J. & Oja, E. Blind Separation from ε-Contaminated Mixtures. Proc. of IX European Signal Processing Conference (EUSIPCO'98), Rhodes, Greece, September 8-11, 1998, vol. III, pp. 1817-1820.

Pajunen, P. Blind Source Separation Using Algorithmic Information Theory. Proc. of Workshop on Independence and Artificial Neural Networks (I&ANN'98), Rhodes, Greece, February 9-10, 1998, pp. 26-31.

Pajunen, P. Blind Source Separation Using Algorithmic Information Theory. Neurocomputing, 1998. Vol. 22, pp. 35-48.

Ypma, A. & Pajunen, P. Second-Order ICA in Machine Vibration Analysis. Helsinki University of Technology, Publications in Computer and Information Science, Report A53, Espoo, Finland, 1998, 54 p.

Peura, M. Object Recognition via Hierarchical Matching. Proc. of XXIII Convention on Radio Science and Remote Sensing Symposium, Espoo, Finland, August 24-25, 1998, pp. 91-92.

Peura, M., Koistinen, J. & King, R. Visual Modelling of Radar Images. Proc. of COST-75 Final International Seminar on Advanced Weather Radar Systems, Locarno, Switzerland, March 23-27, 1998.

Peura, M. The Self-Organizing Map of Trees. Neural Processing Letters, 1998. Vol. 8, pp. 155-162.

Raivio, K., Henriksson, J. & Simula, O. Neural Detection of QAM Signal with Strongly Nonlinear Receiver. Neurocomputing, 1998. Vol. 21, nro 3, pp. 159-171.

Korhonen, A., Cser, L., Larkiola, J., Myllykoski, P., Simula, O. & Ahola, J. Application of Neural Networks for Modelling of Rolling and Annealing of Steel Strip. Proc. of ESAFORM Conference, Sophia-Antipolis, France, March 20-28, 1998, pp. 251-254.

Simula, O., Vesanto, J. & Vasara, P. Analysis of Industrial Systems Using the Self-Organizing Map. Proc. of 1998 Second International Conference on Knowledge-Based Intelligent Engineering Systems (KES'98), Adelaide, Australia, April 21-23, 1998, pp. 61-68.

Cser, L., Korhonen, A., Simula, O., Larkiola, J., Myllykoski, P. & Ahola, J. Knowledge Based Methods in Modelling of Rolling. Proc. of CIRP International Seminar on Intelligent Computation in Manufacturing Engineering, Capri, Italy, July 1-3, 1998, pp. 265-271.

Cser, L., Korhonen, A., Simula, O., Larkiola, J. & Ahola, J. The SOM Based Data Mining in Hot Rolling. Proc. of 4th International Symposium on Measurement Technology and Intelligent Instruments (ISMTII'98), Miskolc, Hungary, September 2-4, 1998.

Särelä, J., Vigário, R., Jousmäki, V., Hari, R. & Oja, E. ICA for the Extraction of Auditory Evoked Fields. Proc. of 4th International Conference on Functional Mapping of the Human Brain (HBM'98), Montreal, Canada, June 7-12, 1998, p. 664.

Tang, H. & Simula, O. Another Dimension of Flow Control for the Intelligent Node. International Conference on Telecommunications (ICT'98), Chalkidiki, Greece, June 22-25, 1998, pp. 425-429.

Valkealahti, K. & Oja, E. Reduced Multidimensional Co-Occurrence Histograms in Texture Classification. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998. Vol. 20, pp. 90-94.

Valkealahti, K. & Oja, E. Texture Classification with Single- and Multiresolution Co-Occurrence Maps. International Journal of Pattern Recognition and Artificial Intelligence, 1998. Vol. 12, pp. 437-452.

Valkealahti, K. & Oja, E. Reduced Multidimensional Histograms in Color Textrure Description. Proc. of 14th International Conference on Pattern Recognition, Brisbane, Australia, August 17-20, 1998, pp. 1057-1061.

Vesanto, J., Himberg, J., Siponen, M. & Simula, O. Enhancing SOM Based Data Visualization. Proc. of 5th International Conference on Soft Computing and Information / Intelligent Systems (IIZUKA'98), Iizuka, Japan, October 16-20, 1998, pp. 64-67.

Vigário, R., Jousmäki, V., Hämäläinen, M., Hari, R. & Oja, E. Independent Component Analysis for Identification of Artifacts in Magnetoencephalographic Recordings. In: Jordan, M.I., Kearns, M.J., & Solla, S.A. (eds.), Advances in Neural Information Processing Systems 10 (NIPS'97, Denver, Colorado, USA, December 2-4, 1997). Cambridge, MA 1998, The MIT Press, pp. 229-235.

Vigário, R., Särelä, J. & Oja, E. Independent Component Analysis in Wave Decomposition of Auditory Evoked Fields. Proc. of International Conference on Artificial Neural Networks, Skövde, Sweden, September 2-4, 1998, pp. 287-292.