Journal of Mathematical Imaging and Vision 2, 261–272 (1992).

Clustering Properties of Hierarchical Self-Organizing Maps

JOUKO LAMPINEN AND ERKKI OJA
Department of Information Technology, Lappeenranta University of Technology, P.O. Box 20, SF-53851 Lappeenranta, Finland

Abstract. A multilayer hierarchical self-organizing map (HSOM) is discussed as an unsupervised clustering method. The HSOM is shown to form arbitrarily complex clusters, in analogy with multilayer feedforward networks. In addition, the HSOM provides a natural measure for the distance of a point from a cluster that weighs all the points belonging to the cluster appropriately. In experiments with both artificial and real data it is demonstrated that the multilayer SOM forms clusters that match better to the desired classes than do direct SOM's, classical k-means, or Isodata algorithms.

Key words: cluster analysis, self-organizing maps, neural networks

1 Introduction

Most of the neural-network research in pattern recognition, image processing, and vision has been focused on supervised learning. Supervised neural networks, such as the multilayer perceptron (MLP), provide a highly efficient model-free method for designing an arbitrarily complex nonlinear classifier iteratively from examples. There are many sample cases showing the power of neural classifiers over classical methods; for a review see, e.g., [1].

A central theoretical result giving impetus to the increasing interest in neural networks is that an MLP with only one nonlinear hidden layer can approximate any continuous function on a compact domain to an arbitrary precision, or as a classifier it can form arbitrarily complex class boundaries [2], [3]. The close relationship between the outputs of the MLP and the optimal Bayes classifier has also been clarified [4].

However, there are some problem domains that cannot be solved with merely a powerful classifier. When the abstraction level of the classification task increases, the shapes of the regions associated together become increasingly complex, requiring impossibly large amounts of training data to form the class boundaries.

Perhaps the most important examples of such problems are in machine vision and image understanding. The essential tasks are locating and recognizing individual objects and compiling a useful interpretation from the objects and their relations. Both of these subtasks have proven to be extremely difficult. The classical approach of programming the *a priori* knowledge or model of the objects into the solution has severe limitations in handling all the natural variations in images. Also, the methods cannot easily adapt to unanticipated or changing situations.

To handle the large variability of natural scenes the image-analysis system must have a large number of free parameters in the early stages, and estimating the parameters requires a lot of data. Using any neural network trained by supervision for the entire image-analysis system would therefore require a huge network with a correspondingly huge number of manually classified samples, and collecting the samples would clearly be very expensive.

This dilemma can be solved by using unsupervised learning techniques in early stages to reduce the number of degrees of freedom in the data. Then the final supervised classifiers, giving semantic labels to objects or their primitives, can have a much smaller number of free parameters, thus requiring fewer preclassified training samples. This scheme is especially suitable for scene analysis, since it is fairly unexpensive to collect large amounts of image data and to train a neu-

ral network with them, as long as the images do not need manual analysis and classification.

The biological neural networks seem to have a similar basic structure, at least at the lowest levels. Although the very first stages in the sensory information pathways are genetically determined, the signals are thereafter fed to sensory maps. It has been shown that these maps can be formed by means of self-organization [5], and there is biological evidence from, e.g., deprivation experiments that the maps are indeed spanned by input data during the development of the network. Very profound conclusions about biological-image-analysis methods cannot be drawn at present. The role of expectations and guessing as means of creating bootstrap data for supervised learning is not known, but it is likely that the biological systems contain continuous hypothesis-generation and testing mechanisms and that feedback throughout the system is a very important factor in guiding the learning process.

In the next sections we discuss the use of a multilayer version of the self-organizing map (SOM) neural network, the hierarchical self-organizing map (HSOM), as a clustering preprocessor in an image-analysis system. The basic one-layer SOM, covered in section 2, divides the input space into convex regions in a fashion analogous to a one-layer feedforward network. The SOM is shown to have more desirable properties than do classical clustering methods: it provides a natural measure for the distance of a point from a cluster that is adaptive to the local statistics of the data. The SOM forms one complex shape following the data distributions in the space, so that regions of the map can be interpreted as clusters in the space, and the difference of the cluster indices is correlated to the weighted distance of all the points in the clusters. In the multilayer hierarchical SOM, discussed in section 3, the outputs of the first SOM are fed into another SOM as input, causing the SOM to divide into distinct cluster representations. The HSOM is shown to form arbitrarily complex clusters, in analogy with multilayer feedforward networks. Section 4 contains experimental results on synthetic and real data that confirm the desirable properties of the HSOM.

2 Self-Organizing Map

2.1 Basic SOM

The SOM, introduced by Kohonen [6], is one of the best-known unsupervised-learning neural networks. It belongs to the class of vector-coding algorithms. In vector coding the problem is to place a fixed number of vectors, the codewords, into the input space, which is usually a high-dimensional real space. Each codeword will correspond to and represent a part of the input space: the set of points in the space that are closer in distance to that codeword than to any other codeword. This produces a Voronoi tessellation into the space. The overall criterion in usual vector coding is to place the codewords in such a way that the average distances from the codewords to the input points belonging to their own Voronoi compartment are minimized.

One way to understand the SOM is to consider it to be a neural-network implementation of this basic idea: each codeword is the weight vector of a neural unit. However, there is an essential extra feature in the SOM. The neurons are arranged in a 1-, 2-, or multidimensional lattice such that each neuron has a set of neighbors, e.g., in two dimensions either four or eight neighbors can be chosen. The goal of learning is not only to find the most representative code vectors for the input space in mean square sense but also to realize a topological mapping from the input space to the grid of neurons. Mathematically, this can be defined as follows.

For any point x in the input space Ω, one or several of the codewords are closest to it. In the following, the distance is the Euclidean distance, but a generalization to other distance functions would be straightforward. Assume that m_b is the closest among all the codewords $m_1, \ldots, m_M$:

$$\| x - m_b \| = \min \| x - m_j \|, \quad j = 1, \ldots, M, \qquad (1)$$

Where j is the usually multidimensional index giving the position of m_j in the lattice. To make the correspondence unique, assume that some

tie-breaking rule is used if several codewords happen to be at exactly the same minimum distance from x. The unit b having the weight vector m_b is then called the *best-matching unit* for vector x, and index $b = b(x)$ can be considered to be the output of the map. Note that for fixed x equation (1) defines the index b of the best-matching unit and for fixed b equation (1) defines the Voronoi compartment of unit b as the set of points that satisfy (1). By the above relation the input space is mapped to the discrete set of neurons. If each neuron is taken to represent one cluster, then the clusters will have a convex polyhedral shape.

A topological mapping is defined as follows: if an arbitrary point $x \in \Omega$ is mapped to unit i, then all points in a neighborhood of x are mapped either to i itself or to one of the units in the neighborhood of i in the lattice. This implies that if i and j are two neighboring units, then their Voronoi compartments have a common boundary. Whether the topological property can hold for all units, however, depends on the dimensionalities of the input space and the neuron lattice. In some earlier works on topologically ordered neuron layers [7], such a mapping was made one-to-one by using a continuum instead of the discrete neuron lattice and by requiring that neighborhoods of points be mapped to neighborhoods. Because no genuine topological maps between two spaces of different dimensions can exist, a two-dimensional neural layer can only follow locally two dimensions of the multidimensional input space.

The Kohonen algorithm for self-organization of the code vectors is as follows [5]:

(i) Choose initial values randomly for the weight vectors m_i of the units i.
(ii) Repeat steps (iii) and (iv) until the algorithm has converged:
(iii) Draw a sample x from the probability distribution of the input samples, and find the best matching unit b according to equation (1).
(iv) Adjust the weight vectors of all units by

$$m_j := m_j + \gamma * h_{b,j} * (x - m_j), \qquad (2)$$

where γ is a gain factor and $h_{b,j}$ is the so-called neighborhood function; usually it is a function of the distance $b-j$ of units b and j measured along the lattice. (In the original version [6] the neighborhood function was equal to 1 for a certain neighborhood of b and was 0 elsewhere. The neighborhood and the gain γ should slowly decrease in time.)

The convergence and the mathematical properties of this algorithm have been considered by several authors, e.g., [6], [8], and [9].

2.2 SOM Optimization

The map algorithm is related to an energy function in [8] and [10]. Let V_b denote the set in the input space where (1) holds, i.e., the Voronoi compartment of unit b. Let $p(x)$ denote the probability density of the inputs x. Define the cost or energy function as

$$E(m_1, \ldots, m_M) = \sum_i \int_{V_i} \sum_k h_{i,k} \| x - m_k \|^2 p(x) d(x). \qquad (3)$$

The functional (3) is piecewise differentiable. Let us write it in the equivalent form

$$E = E(m_1, \ldots, m_M) = \int \sum_k h_{b,k} \| x - m_k \|^2 p(x) d(x), \qquad (4)$$

with b defined appropriately as the index of the best-matching unit. This moves the discontinuity of the V_i to the function

$$b = b(x, m_1, \ldots, m_M). \qquad (5)$$

The usual way to minimize a functional like E, in which the density $p(x)$ is unknown, is to resort to sample functions: for each x define

$$E_1(x, m_1, \ldots, m_M) = \sum_k h_{b,k} \| x - m_k \|^2 . \qquad (6)$$

Functional E is the mean value of this with respect to the density $p(x)$. Functional E_1 is well defined and unique (i.e., a function) almost everywhere in the space of its arguments, except the set of $x, m_1, \ldots, m_M$ defined by the condition

that x has exactly the same distance from two or more points m_i:

$$S = \{x, m_1, \ldots, m_M | \parallel x - m_b \parallel = \parallel x - m_i \parallel \text{ for some } i \neq b\}. \quad (7)$$

In fact, for any fixed $m_1, \ldots, m_M$ the set S consists of all the borders of the Voronoi tessellation in the x-space. Denote the complement of S by $\overline{S}$. In S the index b, hence $h_{b,k}$, is not unique, but in $\overline{S}$ the index b is unique and is piecewise constant. It is not affected by any gradient with respect to x or one of the m_i. Note that S is a closed set and that $\overline{S}$ is open. This means that if $(x, m_1, \ldots, m_M) \in \overline{S}$, there is some ϵ-neighborhood that is also in $\overline{S}$. Any differential change in x or some m_i stays within this neighborhood. Because b cannot change its value within one connected region of S, it follows that b is constant over this neighborhood.

It now holds that E_1 is differentiable in all m_i as long as $(x, m_1, \ldots, m_M) \in \overline{S}$, and it holds that

$$\frac{dE_1}{dm_i} = -2h_{b,i}(x - m_i). \quad (8)$$

A steepest-descent minimization of E_1 leads directly to the usual SOM learning rule:

$$m_i(t+1) = m_i(t) - \frac{1}{2}\gamma \frac{dE_1}{dm_i(t)} \quad (9)$$

$$= m_i(t) + \gamma h_{b,i}(x(t) - m_i(t)) \quad (10)$$

$$i = 1, \ldots, n. \quad (11)$$

Thus the original SOM algorithm (with constant neighborhood function) is a gradient-descent method based on sample functions E_1.

It was shown by Kohonen [10] that when the goal is to minimize the original function E of (3), extra terms appear in the algorithm because of the discontinuities at the set $\overline{S}$.

The minimization of E becomes straightforward in a special case when there is no neighborhood,

$$h_{i,j} = \delta_{i,j}. \quad (12)$$

In this case the learning algorithm and the resulting behavior of the map become similar to vector quantization (VQ) according to the k-means algorithm [11]. This can be seen from the following expansion, which may have wider applicability in the analysis of the energy function E. It can be further expressed in a form that contains only the zeroth-, first-, and second-order moments over regions V_i but no other integrals. Let

$$\omega_i = \int_{V_i} p(x)dx,$$

$$c_i = \frac{1}{\omega_i} \int_{V_i} xp(x)dx,$$

$$\sigma_i = \int_{V_i} \parallel x - c_i \parallel^2 p(x)dx.$$

Then

$$E = \sum_i \sum_k h_{i,k} \int_{V_i} \parallel x - m_k \parallel^2 p(x)dx$$

$$= \sum_i \sum_k h_{i,k} \int_{V_i} [\parallel x - c_i \parallel^2 + \parallel c_i - m_k \parallel^2 - 2(x - c_i)^T(c_i - m_k)]p(x)dx$$

$$= \sum_i \sigma_i + \sum_i \sum_k h_{i,k}\omega_i \parallel c_i - m_k \parallel^2 .$$

It has been assumed that $\sum_k h_{i,k} = 1$ for all i.

If $h_{i,k} = \delta_{i,k}$, then we obtain

$$E = \sum_i \sigma_i + \sum_i \omega_i \parallel c_i - m_i \parallel^2 . \quad (13)$$

Now $m_i = c_i$ is at least a local minimum because at these points the gradient with respect to m_j is zero. At the same time, the sum $\sum_i \sigma_i$ is minimized. This is the basic VQ coding solution.

2.3 SOM and Clustering

Many proposed clustering algorithms have been based on minimum-spanning trees, graph theory, etc. [12] capable of forming arbitrarily complex clusters. The methods use different distance measures $D(x_i, C_k)$ for the points $x_1, \ldots, x_N$ to be clustered from the clusters $C_1, \ldots, C_K$, and they also use different iterative or one-pass algorithms by which all the points are allocated into clusters. A usual criterion is based on the

distance of each point from its nearest cluster: for point x_i, let $D(x_i, C_{k(i)}) = \min_k D(x_i, C_k)$. Then the function to be minimized in clustering is

$$J(C_1, \ldots, C_K) = \sum_{i=1}^{N} D(x_i, C_{k(i)}). \quad (14)$$

If $D(x_i, C_k)$ is measured as the distance of x_i from the nearest point in the cluster C_k,

$$D(x_i, C_k) = \min_j \{d(x_i, y_j), y_j \in C_k\}, \quad (15)$$

the resulting clusters will be long chains in the space, which is desirable if the data from each class are known to have long irregular distributions. The single-linkage clustering method [12] uses this distance measure. However, the distance from the closest point does not take into account the cluster shape. A good measure would be an appropriately weighted distance from all the points in the cluster, assuming that it could be computed without large cost.

The SOM is now shown to use such a measure implicitly. The following analysis is based on the energy function E in (3). For comparisons with the standard clustering framework, we assume in the following that the input distribution is discrete uniform, i.e., there exists only a finite sample $x_1, \ldots, x_N$ of possible input vectors. We also assume that the neighborhood function $h_{b,j}$ is a function of the difference $j - b$ only and use the name $h(j - b)$ for it. The function will be assumed to be spherically symmetrical and monotonically nonincreasing in the sense that

$$h(k_1) = h(k_2) \quad \text{if } \| k_1 \| = \| k_2 \|, \quad (16)$$

$$h(k_1) \geq h(k_2) \quad \text{if } \| k_2 \| \geq \| k_1 \|, \quad (17)$$

where $\| k_1 \|$ is a norm of the discrete (multidimensional) index space.

Instead of the Voronoi compartments, it is now more appropriate to use the following Voronoi index sets:

$$I_b = \{i | \| x_i - m_b \| = \min \| x_i - m_j \|, j = 1, \ldots, M\}, \quad (18)$$

which gives the indices of all vectors x_i falling into the Voronoi compartment of m_b.

The energy function now becomes

$$E(m_1, \ldots, m_M) = \sum_i \sum_k h(i - k) \sum_{p \in I_i} \| x_p - m_k \|^2 . \quad (19)$$

The cost introduced by one data sample x_p is

$$E'(x_p) = \sum_k h(b(x_p) - k) \| x_p - m_k \|^2 . \quad (20)$$

The cost function $E'(x_p)$ can be interpreted as the distance from the point x_p to the cluster represented by the whole SOM network, and learning tries to minimize the total distance from points to the cluster.

When the SOM training has converged, the gradient of the cost function is zero for each unit, regardless of whether the state is a global or a local minimum:

$$\frac{dE}{dm_k} = -2 \sum_i h(i - k) \sum_{p \in I_i} (x_p - m_k) = 0. \quad (21)$$

Note that, because there are only a finite set of vectors x_i, a differential variation of m_k will not change the index sets I_i. Denote now the number of vectors x_p for $p \in I_i$ by N_i, and denote the mean of x_p for $p \in I_i$ by c_i, i.e., $c_i = (1/N_i) \sum_{p \in I_i} x_p$. We now make the approximations

$$N_1 = N_2 = \cdots = N_M, \quad (22)$$

and

$$\sum_i h(i - k) = 1 \quad \text{for all } k. \quad (23)$$

Equation (23) is no restriction because any constant value can be used instead of unity. Equation (22) can be motivated as follows: it is equivalent to the condition that, under the assumption of equal probabilities for all input vectors $x_1, \ldots, x_N$, each unit m_k has an equal probability of being the best-matching unit. In training with the Kohonen algorithm, for high-dimensional data the units will become equiprobable. Because of equation (22), equation (21) gives

$$\sum_i h(i - k)(c_i - m_k) = 0, \quad (24)$$

and, finally, because of (23),

$$m_k = \sum_i h(i-k)c_i. \tag{25}$$

This can be interpreted as a convolution of the sequence h with the sequence $c_1, \ldots, c_M$ to yield the corresponding vectors m_k. Now (25) can be substituted into the cost function (20) to yield

$$E'(x_p) = \sum_k h(b(x_p)-k) \cdot \left\| x_p - \sum_i h(i-k)\frac{1}{N_i}\sum_{j\in I_i} x_j \right\|^2. \tag{26}$$

In (26) the summation is first computed over all the units i and then over all the samples x_j mapped to the unit. By changing the summation to run over all the data samples we get

$$E'(x_p) = \sum_k h(b(x_p)-k) \cdot \left\| x_p - \sum_{r=1}^{N} h(b(x_r)-k)\frac{1}{N_{b(x_r)}} x_r \right\|^2. \tag{27}$$

To further simplify the expression we can approximate the unit locations m_j by the centers of their tessellations c_j since the neighborhood function is a low-pass filter (by (17) it is a non-increasing function), and then each unit will be in the weighted average of the tessellation centers of its neighbors. Clearly, the more curved the map is, the more the m_j move away from the c_j, since a curved map contains more high frequencies that the neighborhood filters out. For a locally linear map low-pass filtering has no effect and the m_j coincide with the c_j. Then Eq. (20) simplifies to

$$E'(x_p) = \sum_k h(b(x_p)-k) \cdot \left\| x_p - \frac{1}{N_k}\sum_{i\in I_k} x_i \right\|^2. \tag{28}$$

The cost introduced by x_p is then the weighted distance from all the other points in the training data. The weighting depends on how far away the points are mapped on the lattice. The virtue of the weighting is that the weighting always encompasses roughly the same amount of data samples and since the distances on the map reflect the distances in the input space, the weighting decreases as the distance of the data points increases.

3 Hierarchical SOM

The hierarchical SOM is here defined as a two-dimensional SOM whose operating principle is as follows:

(i) For each input vector x the best matching unit is chosen from the first-layer map and its index b is input to the second layer.
(ii) The best-matching unit for b is chosen from the second-layer map and its index is the output of the network.

One thing is immediately clear from the above: because each first layer map unit i has a convex polyhedral Voronoi region V_i defined by (1) and each second-layer unit j is the best-matching unit for a subset, say, $i_1, \ldots, i_K$, of the first-layer indices, the second-layer unit is in fact the best matching unit for any $x \in \cup_{k=1}^{K} V_{i_k}$. This region is an arbitrary union of nonoverlapping convex polyhedral regions. Any region in $\Re^n$ can be approximated by such a union to an arbitrary accuracy when the number of component regions V_{i_k} is arbitrarily large. Thus clusters of arbitrary shapes can be represented by the two-layer map.

By analogy to the basic approximation result of two-layer Perceptron networks [3], this is purely an existence result. There is no guarantee that a certain predetermined cluster shape could be learned by the map. However, in unsupervised learning this is an empty question because by definition no target clustering can exist.

In [9] and [13] the theory of hierarchical maps is derived from the principles of coding theory. It is shown that the hierarchical map minimizes the decoding squared error if the training neighborhood in SOM equals the probability distri-

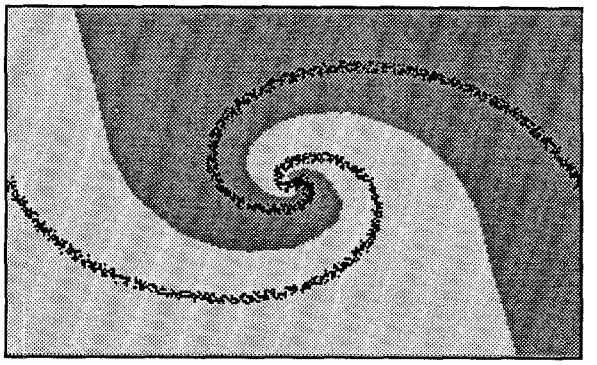

Fig. 1. Example of clusters forms of the HSOM for spiral data.

bution of errors in the codes. In hierarchical VQ the higher-order map quantization error is the source for the lower-order map code errors. In this context the hierarchical SOM is optimal in quantization.

In clustering terminology, whereas the first SOM layer forms one large cluster of all the data samples so that the total distance of the samples from the cluster is minimized, the second map in HSOM then splits the large cluster into equal-size parts. Since the distance relations of the data samples are preserved on the map, the cluster numbers or indices of the best-matching units can be used as a measure of distance of the original data samples. What is gained by the HSOM is that each high-dimensional data vector is mapped to a low-dimensional discrete value so that a comparison of the values implicitly contains a comparison of the original distances.

Figure 1 shows an example of decision regions of the HSOM. The black spiral-shaped stripes are the data points. The HSOM in the figure contained 100 units in the first layer and two units in the second layer. Regions mapped to the two clusters defined by the two second-layer units are shown as different gray levels.

The main advantage of HSOM clustering with respect to classical clustering methods, e.g., k-means, is the adaptive distance measure. In the k-means or Isodata family of methods [11] clusters that are too large are split into smaller ones and clusters that are too small are merged together until all the clusters are of the desired size. In practice it is very difficult to determine a suitable size for the clusters. Also, Isodata clustering algorithms can make only convex clusters because of the nearest-neighbor clustering rule.

Unlike simple linkage clustering, the HSOM offers a distance measure that takes into account all the points in the cluster. As can be seen from (20), the cost introduced by one data point contains the distance of the point from all the other clusters, weighted by the distance of each cluster along the lattice.

4 Experimental Results

In an experiment with artificial data, random Gaussian input classes with elongated shapes were generated (see figure 2) and the confusion matrices of clusterings, i.e., the distribution of each class among the clusters, were formed. The perfect clustering would be such that each cluster contains data points from only one class.

An appropriate measure for the goodness of any clustering ensemble is the mean columnwise entropy of the confusion matrix. It measures the width of the distribution of points from one class. For example, if we have 10 true classes and 40 clusters, the perfect clustering would map each class into a maximum of four clusters, corresponding to the entropy 2.0. An entropy value of 3, say, would indicate that each class is on average distributed into eight clusters.

In each experiment run we generated a new data ensemble of 10 Gaussian-distributed classes with random class centers, principal-axis directions, and variances. The variances of the classes were bounded so that the main-axis variance was randomly 1 to 10 times the minor-axis variance. The number of data samples in each class was either constant ($N = 300$) or random ($200 \leq N \leq 400$).

Clustering the data was tested with 1-dimensional maps of sizes 40, 20, 10, and 5 units for both SOM and HSOM. The first-layer map for the HSOM always contained 160 units.

The results averaged over several class ensembles are presented in figures 3 and 4. The figures show two entropy measures for both SOM and HSOM networks: clusters/class is the column-

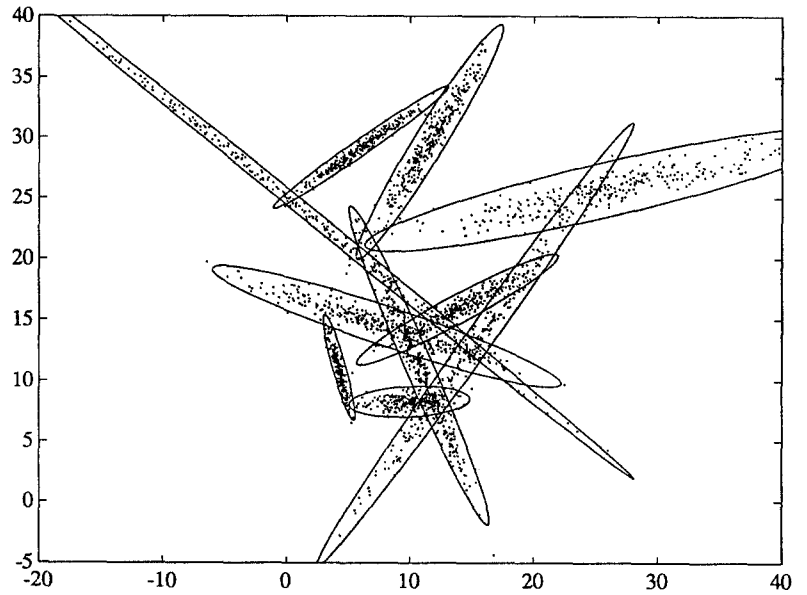

Fig. 2. Example test set of 10 clusters.

wise entropy, i.e., it tells how many clusters are used to cover one class, and classes/cluster is the rowwise entropy, i.e., it tells how many classes each cluster has collected samples from. The ordinate axes in the figures show the widths of the distribution compared to those for perfect clustering. For example, the value 2.0 for (10 classes)/(40 clusters) means that the classes are on average spread into eight clusters instead of four, which would be the optimum case.

In figure 3 all the classes contained the same number of data points ($N = 300$), which favors SOM-type clustering methods that try to make equal-size clusters. In figure 4 the class sizes were random ($200 \leq N \leq 400$), so that optimal VQ of the space is quite different from optimal clustering.

In every case the SOM has a better clusters/class measure, since the clusters of SOM are more compact and fewer clusters are needed to represent a class. The classes/cluster measure is better for HSOM, since the one-layer map cannot track the class boundaries as well as HSOM and each cluster collects points from nearby classes (cf. figure 1). The classification error depends on the classes/cluster measure, since the classifier cannot separate the classes once they have been mapped to the same cluster.

Direct classification errors were also measured for SOM and HSOM, and the results are given in table 1. Each unit was labeled into the class that gave the largest number of hits for the unit, and the classification errors were all the hits from any other classes. This corresponds to the *a posteriori* Bayes classifier, since the hit rates for each cluster measure the probability density of the classes in the cluster regions, and classification is done according to the largest probability.

Note that the class distributions in figure 2 would be rather easy to separate with any simple classifier and that the classification errors of unsupervised clustering cannot be compared with any result of direct-supervised classifiers. The purpose of the clustering network was, as explained in section 1, to reduce the complexity of the data when there were not enough preclassified samples to train a supervised classifier. For the same reason, the SOM network was not fine tuned by LVQ [5]. The classification error gives only an approximate lower limit for the number of errors if only the cluster identities are passed on to the classifier. For comparison, the Bayes classification error, estimated by the one-nearest-neighbor rule by using all the data samples, is about 5%, whereas the errors after

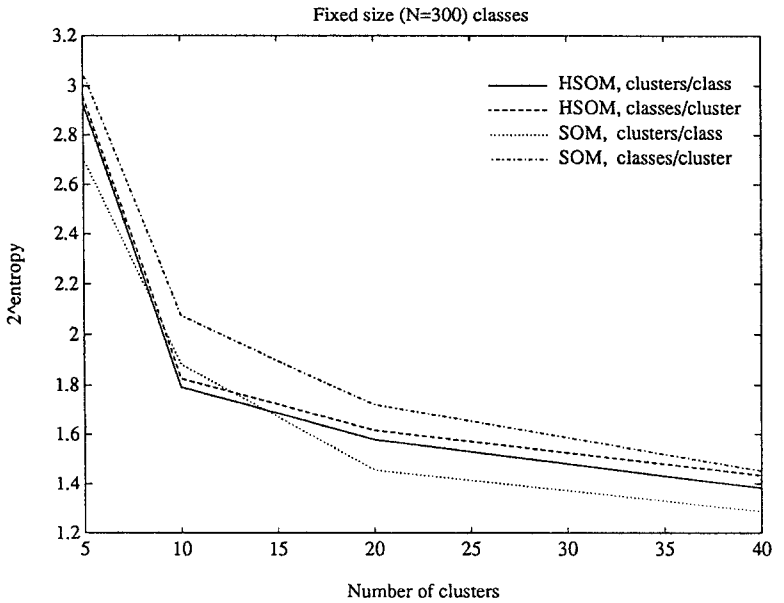

Fig. 3. Clustering entropies for fixed-size clusters.

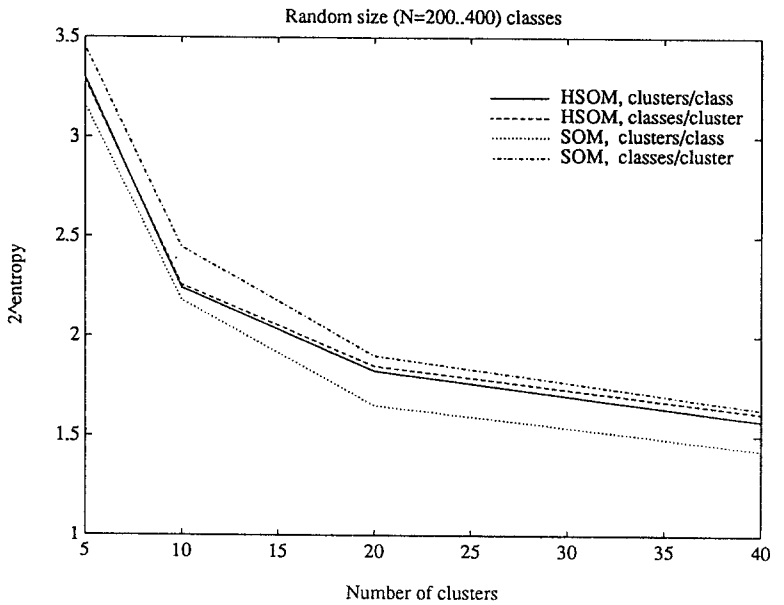

Fig. 4. Clustering entropies for random-size clusters.

the different clustering methods are about 30%.

The SOM and k-means have similar figures for classification errors in Table 1, which stems from the fact that k-means is effectively a batch-training version of the SOM without a neighborhood [9].

The experiments with the Isodata algorithm were performed with the Khoros data-processing and visualization tool[1]. The Isodata algorithm implemented in the Khoros system contains enhanced features to adapt the clusters to the local statistics of the data and to simplify selection of the split and merge parameters, but still it required much more manual experimenting than the almost automatic HSOM to find the optimal clustering.

Table 1. Classification errors in the clustering tests.

Method	Errors(%)
k-means	30.3
SOM	30.7
Isodata	28.0
HSOM	26.3

As a practical example we used HSOM to cluster sensory information from low-level feature detectors in a computer-vision system. The special feature of sensory information is that the signal space is often very high dimensional, but the actually occurring signals are implicitly rather low dimensional since the primary feature detectors tend to be orthogonal or otherwise mutually exclusive.

As the primary feature detectors we used Gabor filters (Gaussian band-pass filters in the frequency domain) in eight different orientations and two frequencies, $\pi/2$ and $\pi/4$, giving a total of 16 different spatial filters, which were applied to each point in the image. The ensuing 16-dimensional feature vectors at each pixel position were mapped with a 100-unit SOM to feature values. Another set of features were obtained by first mapping both resolutions separately by a 100-unit map and then clustering the map outputs by a second 100-unit map. For a complete description of the object-recognition system see [14].

[1] Khoros is software environment for data processing and visualization, a free software package copyrighted by the University of New Mexico, Albuguerque, NM 87106.

Fig. 5. Test images for table 2. from left to right and from top to bottom: T1, T2, J1, J2, P1, P2.

Distortions in the imaging (i.e., different lightings, contrasts, viewing angles) make the Gabor-filter responses move in the 16-dimensional space in a regular way. For example, increasing contrast makes all the edges sharper, increasing the responses of the high-frequency filters. If the clustering algorithm can find such regular trajectories and map them to the same cluster, the distortion tolerance of classification should be increased.

As a test problem we recognized human faces. According to our experiments the feature set produced by the HSOM is indeed clearly more distortion tolerant than that of the direct SOM. Examples of the tests are given in table 2. We compared the features by compiling normalized histograms of the features over each image in figure 5 and computing inner products of the histograms; this is a simplified form of the subspace classifier. One of the test images from each class was selected as the class prototype, and the other images were compared with the prototype.

In larger experiments we have been able to classify 19 out of 20 similar face images by

using only the histograms of the 100 hierarchical features and the subspace classifier.

Table 2. Comparison of SOM and HSOM features (see text for details).

	Class Prototype Image					
	HSOM			SOM		
Test Image	T1	J1	P1	T1	J1	P1
T1	1.000	0.212	0.212	1.000	0.515	0.627
T2	0.550	0.295	0.508	0.790	0.504	0.698
J1	0.212	1.000	0.730	0.515	1.000	0.933
J2	0.291	0.814	0.717	0.572	0.984	0.980
P1	0.212	0.730	1.000	0.627	0.933	1.000
P2	0.231	0.703	0.860	0.568	0.985	0.971

Face recognition was selected as a test case because human faces clearly contain distinct features that characterize the face invariantly to changes in imaging conditions. The purpose was to verify whether the HSOM network can find such features in an unsupervised way. The classifier used in the experiments is very primitive since it loses all location information from the features and only the relative frequencies of the features in the image are considered. The successful tests show that the features extracted by the HSOM are rather robust and distortion tolerant.

References

1. R. Hecht-Nielsen, "Theory of backpropagation neural network," in *Proc. IEEE Int. Joint Conf. on Neural Networks*, vol. I, Washington, DC, 1989, pp. 593–611.
2. R. Hecht-Nielsen, *Neurocomputing*, Addison-Wesley: Reading, MA, 1990.
3. K. Hornik, M. Stinchcombe, and H. White, "Multi-layer feedforward networks are universal approximators," *Neural Net.*, vol. 2, pp. 359–366, 1989.
4. M.D. Richard and R.P. Lippmann, "Neural network classifiers estimate Bayesian *a posteriori* probabilities," *Neural Comput.*, vol. 3, 1991, pp. 461–483.
5. T. Kohonen, *Self-Organization and Associative Memory*, Springer-Verlag: Berlin, 1989.
6. T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biol. Cybernet.*, vol. 43, pp. 59–69, 1982.
7. S. Amari, "Topographic organization of nerve fields," *Bull. Math. Biol.*, vol. 42, pp. 339–364, 1980.
8. H. Ritter and K. Schulten, "Kohonen's self-organizing maps: exploring their computational capabilities," in *Proc. IEEE Int. Joint Conf. on Neural Networks,* vol. 1, San Diego, CA, 1988, pp. 109–116.
9. S.P. Luttrell, "Self-organisation: A derivation from first principles of a class of learning algorithms," *Proc. IEEE Int. Joint Conf. on Neural Networks*, vol. 2, Washington, DC, 1989, pp. 495–498.
10. T. Kohonen, "Self-organizing maps: optimization approaches," in *Artificial Neural Networks*, vol. 2, T. Kohonen, K. Mäkisara, J. Kangas, and O. Simula, eds., North-Holland: Amsterdam, 1991, pp. 981–990.
11. P.A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*, Prentice Hall: London, 1982.
12. J.C Gower and G.J.S Ross, "Minimum spanning trees and single linkage cluster analysis," *Appl. Statist.*, vol. 18, pp. 54–64, 1969.
13. S.P. Luttrell, "Image compression using a multilayer neural network," *Pattern Recog. Lett.*, vol. 10, pp. 1–7, 1989.
14. J. Lampinen, "Distortion tolerant pattern recognition using invariant transformations and hierarchical SOFM clustering," in *Artificial Neural Networks*, vol. 1, T. Kohonen, K. Mäkisara, J. Kangas, and O. Simula, eds., North-Holland: Amsterdam, 1991, pp. 99–104.

Jouko Lampinen received the M.Sc. degree in applied physics and electronics from the University of Kuopio, Finland, in 1988. Currently he is finishing his doctoral thesis in computer science at Lappeenranta University of Technology, Finland, in the Department of Information Technology. His research interests are pattern recognition and neural networks, especially self-organizing models in feature extraction.

Erkki Oja is Professor of Computer Science at Lappeenranta University of Technology, Finland. He received his M.Sc. and Dr. Tech degrees from Helsinki University of Technology 1972 and 1977, respectively. He was visiting scientist at Brown University, Providence, in 1977-1978 and at Tokyo Institute of Technology in 1983-1984 where he also held the Toshiba Visiting Professor's Chair during the academic year 1990-1991. Professor Oja is the author of a number of articles on pattern recognition, computer vision, and neural computing and is the author of the book Subspace Methods of Pattern Recognition, which has been translated into Chinese and Japanese. His present research interests are in applying neural networks to computer vision and the study of subspace and PCA networks. Professor Oja is member of ACM, IEEE, INNS, ENNS and the Finnish Academy of Sciences, and he serves on the editorial boards of International Journal of Neural Systems, Neural Networks, Neural Computation, and IEEE Transactions on Neural Networks.