Clustering of the Self-Organizing Map

Juha Vesanto and Esa Alhoniemi, *Student Member, IEEE*

Abstract— **The Self-Organizing Map (SOM) is an excellent tool in exploratory phase of data mining. It projects input space on prototypes of a low-dimensional regular grid which can be effectively utilized to visualize and explore properties of the data. When the number of SOM units is large, to facilitate quantitative analysis of the map and the data, similar units need to be grouped, i.e., clustered. In this paper, different approaches to clustering of the SOM are considered. In particular, the use of hierarchical agglomerative clustering and partitive clustering using k-means are investigated. The two-stage procedure — first using SOM to produce the prototypes which are then clustered in the second stage — is found to perform well when compared to direct clustering of the data and to reduce the computation time.**

Keywords— **self-organizing map, data mining, exploratory data analysis, clustering**

I. Introduction

DATA mining process can be divided to six sequential, iterative steps: problem definition, data acquisition, data preprocessing and survey, data modeling, evaluation and knowledge deployment. Each step is essential: the problem defines what data are used and what is a good solution. Modeling makes it possible to apply the results to new data. On the other hand, data modeling without good understanding and careful preparation of the data leads to problems. Finally, the whole mining process is meaningless if the new knowledge will not be used [1].

The purpose of survey is to gain insight into the data — possibilities and problems — to determine whether the data are sufficient and to select the proper preprocessing and modeling tools. Typically, several different data sets and preprocessing strategies need to be considered. For this reason, efficient visualizations and summaries are essential. In this article, we focus on clusters, since they are important characterizations of data.

The Self-Organizing Map (SOM) [2] is especially suitable for data survey because it has prominent visualization properties. It creates a set of prototype vectors representing the data set and carries out a topology preserving projection of the prototypes from the d-dimensional input space onto a low-dimensional grid. This ordered grid can be used as a convenient visualization surface for showing different features of the SOM (and thus of the data), for example the cluster structure [3].

However, the visualizations can only be used to obtain qualitative information. To produce summaries — quantitative descriptions of data properties — interesting groups of map units must be selected from the SOM. The most obvious such a group is the whole map. While its properties are certainly interesting, even more useful summaries can be prepared if the SOM (and thus the data) actually consist of two or more separate regions, and these are studied separately. Another option would be to consider all map units individually, but in the case of a large map this could result in far too many summaries. Thus, to be able to effectively utilize the information provided by the SOM, methods to give good candidates for map unit clusters or groups are required. It should be emphasized that the goal here is not to find an optimal clustering for the data, but to get good insight into the cluster structure of the data for data mining purposes. Therefore, the clustering method should be fast, robust and visually efficient.

The clustering is carried out using a two-level approach, where the data set is first clustered using the SOM, and then the SOM is clustered. The most important benefit of this procedure is that computational load decreases considerably, making it possible to cluster large data sets and to consider several different preprocessing strategies in a limited time. Naturally, the approach is valid only if the clusters found using the SOM are similar to those of the original data. In the experiments, a comparison between the results of direct clustering of data and clustering of the prototype vectors of the SOM is performed and the correspondence is found to be acceptable.

II. Clustering

A. Definitions

A clustering $\mathbf{Q}$ means partitioning a data set into a set of clusters Q_i, $i = 1, \ldots, C$. In crisp clustering, each data sample belongs to exactly one cluster. Fuzzy clustering [4] is a generalization of crisp clustering where each sample has a varying degree of membership in all clusters. Clustering can also be based on mixture models [5]. In this approach, the data are assumed to be generated by several parametrized distributions, typically Gaussians. Distribution parameters are estimated using for example Expectation-Maximation algorithm. Data points are assigned to different clusters based on their probabilities in the distributions. However, neither fuzzy clustering nor mixture models based clustering are considered in this study because the goal was to evaluate clustering of the SOM using a few simple standard methods.

A widely adopted definition of optimal clustering is a partitioning that minimizes distances within and maximizes distances between clusters. However, this leaves much room for variation: within- and between-clusters distances can be defined in several ways, see Table I. The selection of the distance criterion depends on application. The distance norm $||\cdot||$ is yet another parameter to consider. In this paper, Euclidean norm is used because it is widely used with SOM. In addition, the k-means error criterion is based on it.

Many algorithms utilize local criteria in clustering data. For example, S_{nn} and d_s in Table I are based on distance

TABLE I

WITHIN-CLUSTER DISTANCES $S(Q_k)$ AND BETWEEN-CLUSTERS DISTANCES $d(Q_k, Q_l)$; $\mathbf{x}_i, \mathbf{x}_{i'} \in Q_k$, $i \neq i'$, $\mathbf{x}_j \in Q_l$, $k \neq l$. N_k IS THE NUMBER OF SAMPLES IN CLUSTER Q_k AND $c_k = \frac{1}{N_k}\sum_{\mathbf{x}_i \in Q_k} \mathbf{x}_i$.

Within-cluster distance	$S(Q_k)$
average distance	$S_a = \frac{\sum_{i,i'} \|\|\mathbf{x}_i - \mathbf{x}_{i'}\|\|}{N_k(N_k-1)}$
nearest neighbor distance	$S_{nn} = \frac{\sum_i \min_{i'}\{\|\|\mathbf{x}_i - \mathbf{x}_{i'}\|\|\}}{N_k}$
centroid distance	$S_c = \frac{\sum_i \|\|\mathbf{x}_i - \mathbf{c}_k\|\|}{N_k}$

Between-clusters distance	$d(Q_k, Q_l)$
single linkage	$d_s = \min_{i,j}\{\|\|\mathbf{x}_i - \mathbf{x}_j\|\|\}$
complete linkage	$d_{co} = \max_{i,j}\{\|\|\mathbf{x}_i - \mathbf{x}_j\|\|\}$
average linkage	$d_a = \frac{\sum_{i,j} \|\|\mathbf{x}_i - \mathbf{x}_j\|\|}{N_k N_l}$
centroid linkage	$d_{ce} = \|\|\mathbf{c}_k - \mathbf{c}_l\|\|$

to nearest neighbor. However, the problem is that they are sensitive to noise and outliers. Addition of a single sample to a cluster can radically change the distances [6]. To be more robust, the local criterion should depend on collective features of a local data set [7]. Solutions include using more than one neighbor [8], or a weighted sum of all distances. It has been shown that the SOM algorithm implicitly uses such a measure [9].

B. Algorithms

The two main ways to cluster data — make the partitioning — are hierarchical and partitive approaches. The hierarchical methods can be further divided to agglomerative and divisive algorithms, corresponding to bottom-up and top-down strategies, to build a hierarchical clustering tree. Of these agglomerative algorithms are more commonly used than the divisive methods, and are considered in this paper.

Agglomerative clustering algorithms usually have the following steps:

1. initialize: assign each vector to its own cluster
2. compute distances between all clusters
3. merge the two clusters that are closest to each other
4. return to step 2 until there is only one cluster left

In other words, data points are merged together to form a clustering tree which finally consists of a single cluster, the whole data set. The clustering tree, dendrogram, can be utilized in interpretation of the data structure and determination of the number of clusters.

However, the dendrogram does not provide a unique clustering. Rather, a partitioning can be achieved by cutting the dendrogram at certain level(s), see Fig. 1. The most typical solution is to cut the dendrogram where there is a large distance between two merged clusters. Unfortunately this ignores the fact that the within-cluster distance may be different for different clusters. In fact, some clusters may be composed of several subclusters: to obtain sensible partitioning of the data, the dendrogram may have to be cut at different levels for each branch [10]. For example, two alternative ways to get three clusters are shown in Fig. 1.

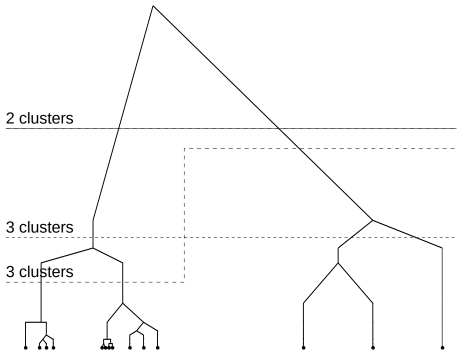


Fig. 1. A dendrogram of a set of 14 points in one-dimensional space. A partitioning can be obtained by cutting the dendrogram at a certain level, for example at the level where there are only two clusters left, because there is a large jump in the dendrogram. An alternative way is to cut the dendrogram at different level for each branch.

Partitive clustering algorithms divide a data set into a number of clusters, typically by trying to minimize some criterion or error function. The number of clusters is usually predefined, but it can also be part of the error function [11]. The algorithm consists of the following steps:

1. determine the number of clusters
2. initialize the cluster centers
3. compute partitioning for data
4. compute (update) cluster centers
5. if the partitioning is unchanged (or the algorithm has converged), stop; otherwise return to step 3

If the number of clusters is unknown, the partitive algorithm can be repeated for a set of different number of clusters, typically from two to $\sqrt{N}$, where N is the number of samples in the data set. An example of a commonly used partitive algorithm is the k-means, which minimizes error function

$$E = \sum_{k=1}^{C} \sum_{\mathbf{x} \in Q_k} \|\mathbf{x} - \mathbf{c}_k\|^2, \tag{1}$$

where C is the number of clusters and $\mathbf{c}_k$ is the center of cluster k.

Partitive methods are better than hierarchical ones in the sense that they do not depend on previously found clusters. On the other hand, partitive methods make implicit assumptions on the form of clusters. For example, k-means tries to find spherical clusters.

To select the best one among different partitionings, each of these can be evaluated using some kind of validity index. Several indices have been proposed [6], [12]. In our simulations, we used Davies-Bouldin index [13], which uses

S_c for within-cluster distance and d_{ce} for between clusters distance. According to Davies-Bouldin validity index, the best clustering minimizes

$$\frac{1}{C}\sum_{k=1}^{C}\max_{l\neq k}\left\{\frac{S_c(Q_k)+S_c(Q_l)}{d_{ce}(Q_k,Q_l)}\right\}, \qquad (2)$$

where C is the number of clusters. The Davies-Bouldin index is suitable for evaluation of k-means partitioning, because it gives low values indicating good clustering results for spherical clusters.

Many clustering methods, for example agglomerative clustering using the complete linkage d_{co}, require the clusters to be compact and well-separated. In real data this is rarely the case. Rather, the gaps between clusters are obscured by noise, the clusters overlap, and there are outliers. When the data set size increases, also computational load becomes a problem, especially if some of the distance measures or indices are computationally expensive, like average distance S_a and average linkage d_a.

C. Two-level approach

The approach used in this paper, clustering of the SOM rather than clustering the data directly, is depicted in Fig. 2. First, a large set of prototypes — much larger than the expected number of clusters — is formed using the SOM or some vector quantization algorithm. The prototypes can be interpreted as "protoclusters" which are in the next step combined to form the actual clusters. Each data vector of the original data set belongs to the same cluster as its nearest prototype. Similar multiple-level approaches to clustering have been proposed earlier, e.g. in [9]. While extra abstraction levels yield higher distortion, they also effectively reduce the complexity of the reconstruction task [14].

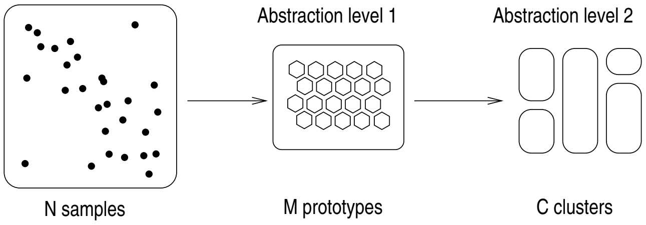


Fig. 2. First abstraction level is obtained by creating a set of prototype vectors using, e.g., the SOM. Clustering of the SOM creates the second abstraction level.

The primary benefit of the two-level approach is the reduction of the computational cost. Even with relatively small number of samples many clustering algorithms — especially hierarchical ones — become intractably heavy. For this reason, it is convenient to cluster a set of prototypes rather than directly the data [15].

Consider clustering N samples using k-means. This involves making several clustering trials with different values for k. The computational complexity is proportional to $\sum_{k=2}^{C_{max}} Nk$, where C_{max} is preselected maximum number of clusters. When a set of prototypes is used as an intermediate step, the total complexity is proportional to $NM + \sum_k Mk$, where M is the number of prototypes. With $C_{max} = \sqrt{N}$ and $M = 5\sqrt{N}$, the reduction of computational load is about $\sqrt{N}/15$, or about six-fold for $N = 10000$. Of course, this is a very rough estimate, since many practical considerations are ignored.

The reduction is even greater for agglomerative algorithms, since they cannot start from $\sqrt{N}$, but have to start with N clusters and work their way down from there. If agglomerative clustering is used during the second step, the two-level strategy is a way to use partitive clustering to avoid those $N - 5\sqrt{N}$ first steps. Table III shows computation times for both direct and two-level clustering strategies in our experiments. Even with these relatively small data sets, the difference is remarkable especially in the agglomerative methods.

Another benefit is noise reduction. The prototypes are local averages of the data and therefore less sensitive to random variations than the original data. In [16] partitive methods (i.e., a small SOM) greatly outperformed hierarchical methods in clustering imperfect data. Outliers are less of a problem since — by definition — there are very few outlier points and therefore their impact on the vector quantization result is limited. On the other hand, this may also be a problem if the outliers are actually the interesting areas. A recently proposed clustering method Chameleon also utilizes a two-level strategy to get more reliable estimates of cluster similarities [8].

III. Clustering of the SOM

A. SOM Training

The SOM consists of a regular, usually two-dimensional, grid of map units. Each unit i is represented by a prototype vector, $\mathbf{m}_i = [m_{i1}, \ldots, m_{id}]$ where d is input vector dimension. The units are connected to adjacent ones by a neighborhood relation. The number of map units, which typically varies from a few dozen up to several thousand, determines the accuracy and generalization capability of the SOM. During training, the SOM forms an elastic net that folds onto the "cloud" formed by the input data. Data points lying near each other in the input space are mapped onto nearby map units. Thus, the SOM can be interpreted as a topology preserving mapping from input space onto the two-dimensional grid of map units.

The SOM is trained iteratively. At each training step, a sample vector $\mathbf{x}$ is randomly chosen from the input data set. Distances between $\mathbf{x}$ and all the prototype vectors are computed. The best-matching unit (BMU), denoted here by b, is the map unit with prototype closest to $\mathbf{x}$:

$$||\mathbf{x} - \mathbf{m}_b|| = \min_i\{||\mathbf{x} - \mathbf{m}_i||\}. \qquad (3)$$

Next, the prototype vectors are updated. The BMU and its topological neighbors are moved closer to the input vector in the input space. The update rule for the prototype vector of unit i is:

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \alpha(t)h_{bi}(t)[\mathbf{x} - \mathbf{m}_i(t)], \qquad (4)$$

where t denotes time, $\alpha(t)$ is learning rate and $h_{bi}(t)$ is neighborhood kernel centered on the winner unit:

$$h_{bi}(t) = \exp\left(-\frac{\|\mathbf{r}_b - \mathbf{r}_i\|^2}{2\sigma^2(t)}\right), \qquad (5)$$

where $\mathbf{r}_b$ and $\mathbf{r}_i$ are positions of neurons b and i on the SOM grid. Both $\alpha(t)$ and $\sigma(t)$ decrease monotonically with time. There is also a batch version of the algorithm where the adaptation coefficient is not used [2].

In the case of a discrete data set and fixed neighborhood kernel, the error function of SOM can be shown to be [17]

$$E = \sum_{i=1}^{N}\sum_{j=1}^{M} h_{bj}||\mathbf{x}_i - \mathbf{m}_j||^2, \qquad (6)$$

where N is number of training samples and M is the number of map units. Neighborhood kernel h_{bj} is centered at unit b, the BMU of vector $\mathbf{x}_i$, and evaluated for unit j. If neighborhood kernel value is one for the BMU and zero elsewhere, this leads to minimization of Eq. 1 — the SOM reduces to adaptive k-means algorithm [18]. If this is not the case, from Eq. 6 it follows that the prototype vectors are not in the centroids of their Voronoi sets, but are local averages of all vectors in the data set weighted by neighborhood function values.

If desired, some vector quantization algorithm, e.g. k-means, can be used instead of SOM in creating the first abstraction level. Other possibilities include:

- Minimum spanning tree SOM [19], neural gas [20], growing cell structures [21] and competing SOMs [22] are examples of algorithms where the neighborhood relations are much more flexible and/or the low-dimensional output grid has been discarded. Their visualization is much less straightforward than that of the SOM.
- Also several such growing variants of the SOM have been proposed where the new nodes do have a well-defined place on low-dimensional grid, and thus the visualization would not be very problematic [23], [24], [25], [26], [27].

The SOM variants were not used in this study, because we wanted to select the most commonly used version of the SOM. However, the principles presented in this paper could naturally be applied to the prototype vectors of these variants as well.

The two-level approach in clustering can only work if the prototypes reflect the properties of the data. The clusters found using the prototypes should be similar to those that would have been found directly from the data. In vector quantization, it has been shown that the density of the prototype vectors is proportional to const $\cdot\, p(\mathbf{x})^{\frac{d}{d+r}}$, where $p(\mathbf{x})$ is the probability density function (pdf) of the input data, d is dimension and r is distance norm [28], [29]. For the SOM, connection between the prototypes and the pdf of the input data has not been derived in general case. However, a similar power law has been derived in one-dimensional case [30]. Even though the law holds only when the number of prototypes approaches infinity and neighborhood width is very large, numerical experiments have shown that the computational results are relatively accurate even for a small number of prototypes [31]. Based on close relation between the SOM and k-means, it can be assumed that the SOM roughly follows the density of training data when not only the number of map units but also final neighborhood width are small as was the case in our experiments.

Finally, the SOM algorithm is applicable to large data sets. The computational complexity scales linearly with the number of data samples, it does not require huge amounts of memory — basically just the prototype vectors and the current training vector — and can be implemented both in a neural, on-line learning manner as well as parallelized [32]. On the other hand, the complexity scales quadratively with the number of map units. Thus, training huge maps is time-consuming, although the process can be speeded up with special techniques, see for example [33], [34]. For example, in [33], a SOM with million units was trained with 6.8 million 500-dimensional data vectors.

B. Visual inspection

An initial idea of the number of clusters in the SOM as well as their spatial relationships is usually acquired by visual inspection of the map. The most widely used methods for visualizing the cluster structure of the SOM are distance matrix techniques [35], [36], especially the unified distance matrix (U-matrix). The U-matrix shows distances between prototype vectors of neighboring map units. Because they typically have similar prototype vectors, U-matrix is actually closely related to the single linkage measure. It can be efficiently visualized using gray shade [37], see for example Figs. 7(a), 11(a), and 12(a).

Another visualization method is to display the number of hits in each map unit. Training of the SOM positions interpolating map units between clusters and thus obscures cluster borders. The Voronoi sets of such map units have very few samples ("hits") or may even be empty. This information can be utilized in clustering the SOM by using zero-hit units to indicate cluster borders [38].

Generic vector projection methods can also be used. As opposed to the methods above, these are generally applicable to any set of vectors, for example the original data set. The high-dimensional vectors are projected to a low dimension so that intersample distances are preserved as faithfully as possible. Thus, the high-dimensional data set can be visualized while still preserving its essential topological properties. Examples of such nonlinear projection methods include multi-dimensional scaling techniques [39], [40], Sammon's mapping [41] and curvilinear component analysis [42]. A special technique is to project the prototype vectors into a color space so that similar map units are assigned similar colors [43], [44].

Of course, based on the visualization, one can select clusters manually. However, this is a tedious process and nothing guarantees that the manual selection is done consistently. Instead, automated methods are needed.

C. SOM clustering

In agglomerative clustering, the SOM neighborhood relation can be used to constrain the possible merges in the construction of the dendrogram [45]. Also, knowledge of interpolating units can be utilized both in agglomerative and partitive clustering by excluding them from the analysis. If this is used together with the neighborhood constraint in agglomerative clustering, the interpolative units form borders on the map which the construction of the dendrogram must obey. It may even be that the interpolating units completely separate some areas from the rest of the map.

The dendrogram of a map with M units contains $M-1$ different non-singleton clusters, which is far too many to facilitate summarization. Additionally, most of these clusters are completely uninteresting: many of them differ only by addition of a single or a few map units. Therefore, the dendrogram has to be pruned. The objective is to find the interesting groups. This can be done by testing for each merge of the dendrogram whether the involved clusters are sufficiently different from each other. If they are, both are added to the list of interesting groups. Interesting merges can be defined for example as follows (see Fig. 3):

- The Davies-Bouldin index (Eq. 2) can be calculated for the two clusters. If the index is greater than one, there is an interesting merge.
- If the gap between the two clusters $d_s(Q_k, Q_l)$ is greater than the sum of mean distances between points in the two clusters $S_{nn}(Q_k) + S_{nn}(Q_l)$, there is an interesting merge.

To be able to evaluate the criteria above, all singleton clusters have to be ignored.

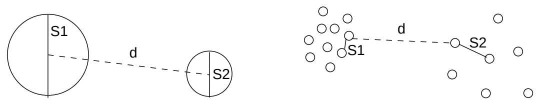


Fig. 3. Different criteria for finding interesting cluster merges: Davies-Bouldin (left) and gap (right). Merge is interesting when $d > S1 + S2$, where $S1$ and $S2$ are within-cluster distances and d is the distance between clusters.

Note that the distance measures used in the gap criterion are very sensitive to noise, especially if the number of samples in a cluster is small. However, the problem can be alleviated somewhat by requiring a wider gap if the sample size is small. Let x be a random sample of size N, whose mean can be estimated as $\mu = E\{x\} = \sum x/N$. It is easy to show that its variance is $Var\{\mu\} = Var\{x\}/N$. In many distribution models, for example the gamma distribution, the variance is directly proportional to expectation $Var\{x\} = a_1 E\{x\}$. Therefore let $Var\{\mu\} = a_1 E\{x\}/N$. Now, let x be the distance between each sample and its nearest neighbor in cluster Q_k of size N_k samples. A robust upper limit for nearest neighbor distance $S_{nn}(Q_k)$ can be acquired by: $S^*_{nn}(Q_k) = E\{x\} + a_2 Var\{x\} = E\{x\} + a_2 a_1 E\{x\}/N_k = E\{x\}(1 + a/N_k)$. In order to make also the between-clusters distance $d_s(Q_k, Q_l)$ more robust, the mean of four shortest distances between the samples can be used instead of only the shortest distance.

The pruning produces a more clear dendrogram, but still does not provide a unique partitioning. This is not a problem if the objective is only to find a set of interesting groups. If necessary, a final partitioning can be selected by hand using some kind of interactive tool [10]. By viewing the clustering hierarchy down from top, one can concentrate on the big, firmly separated clusters and discard the low-level groups. In our experiments, though, an automated procedure for finding the partitioning from the dendrogram was used, see Sec. IV-C for details.

IV. Experiments

A. Data sets

In the experiments, one real-world and two artificial data sets were used. Plots of the three data sets are presented in Fig. 4.

- Data set I ("clown") consisted of 2220 two-dimensional samples. The data set had the following seven clusters:
  - a cluster with three subclusters (right eye),
  - a spherical cluster (left eye),
  - an elliptical cluster (nose),
  - a non-spherical cluster (U-shaped: mouth), and
  - a large and sparse cluster (body).

In addition, some outliers were added to see how they affect the clustering. As it happens, the outliers are mostly concentrated on the left, and seem to form a cluster of their own.

- Data set II was generated to see how the algorithms perform when there is only one cluster: a spherical Gaussian distribution in eight dimensions (1000 samples).
- Data set III consisted of 4205 75-dimensional samples of the technology of pulp and paper mills of the world [46].

Especially in data set III, the original variables had very different scales. Therefore, as part of preprocessing, all variables in each data set were linearly scaled to have zero mean and unit variance.

B. SOMs

A SOM was trained using the sequential training algorithm for each data set [1]. All maps were linearly initialized in the subspace spanned by the two eigenvectors with greatest eigenvalues computed from the training data. The maps were trained in two phases: a rough training with large initial neighborhood width and learning rate and a fine-tuning phase with small initial neighborhood width and learning rate, see Table II. The neighborhood width decreased linearly to 1; neighborhood function was Gaussian (see Eq. 5). The training length of the two phases were 3 and 10 epochs and the initial learning rates 0.5 and 0.05, respectively. The learning rate decreased linearly to zero during the training.

The prototype vectors of the SOMs have been superimposed on the plots of the data sets in Fig. 4. Note that the prototype vectors are well within the data sets, and that

[1] In the SOM training, freely available Matlab package SOM Toolbox was used. For more information, see URL `http://www.cis.hut.fi/projects/somtoolbox/`.

TABLE II

TRAINING PARAMETERS OF THE SOMS. THE PARAMETERS $\sigma_1(0)$ AND $\sigma_2(0)$ ARE INITIAL NEIGHBORHOOD WIDTHS FOR THE ROUGH AND FINE-TUNING PHASES, RESPECTIVELY. THE LAST COLUMN CONTAINS THE TOTAL TRAINING TIME.

data set	map size	$\sigma_1(0)$	$\sigma_2(0)$	elapsed time
I	19×17	10	2	45 s
II	16×13	8	2	20 s
III	24×19	12	2	569 s

this has removed most of the effect of outliers in data set I (Fig. 4(a)).

C. Clustering

For each SOM, a visual inspection was performed. Then, agglomerative and partitive clustering algorithms were applied. Map units with empty Voronoi sets were excluded from the analysis. The neighborhood relations were not used.

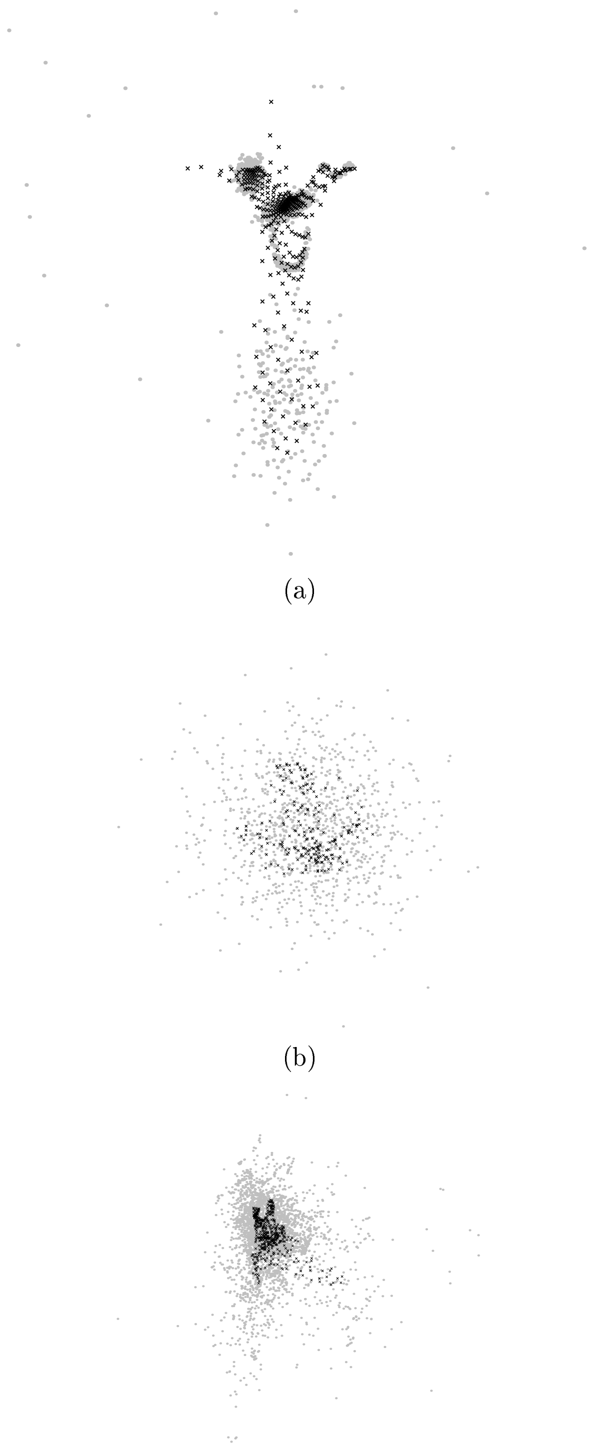
In the agglomerative clustering, single, average and complete linkages were used in the construction phase. The pruning was carried out using gap criterion with $a = 3$ (see Section III-C). Excluding the interpolating units proved to be very important: if this was not done, the interpolating units blurred out the gaps that the criterion was trying to detect.

As mentioned earlier, the list of interesting clusters does not give a specific partitioning. Looking at the pruned list down from top, the clusters form a hierarchy where each high-level cluster consists of some low-level clusters and a set of separate samples, see Fig. 5. The final partitioning was done by the following procedure:

1. A parent cluster was used if the subclusters represented less than half of its samples. Otherwise, the parent cluster was ignored and the subclusters were used instead.
2. All remaining units (interpolative as well as those ignored above) were partitioned to nearest cluster.
3. Each sample in the original data was partitioned to the same cluster as its BMU.

Applying the same agglomerative clustering procedure directly to the data proved to be problematic. Because the original data sets were both much bigger and more noisy than the SOM prototypes, using $a = 3$ resulted in hundreds of small clusters. For data set I, $a = 30$ gave reasonable results and was used for comparison. For data sets II and III, the gap criterion did not work at all. Instead, the comparisons in Section IV-G were done based on partitionings done by simply cutting the dendrograms at certain level.

The partitive clustering of both SOMs and the data sets was carried out using batch k-means algorithm. Because k-means is sensitive to initialization it was run 100 times for each k with different random initializations. The best partitioning for each number of clusters was selected using error criterion in Eq. 1. Another possibility would have been to use some annealing technique to better avoid local minima of the error function [47], [48], [49].



(a)

(b)

(c)

Fig. 4. Illustration of data set I (a), II (b), and III (c). Two-dimensional data set I is directly plotted in two dimensions, whereas data sets II and III are projected to the two-dimensional subspace spanned by the two eigenvectors with largest eigenvalues. The gray dots are data points and the black crosses are prototype vectors of the SOM trained with corresponding data.

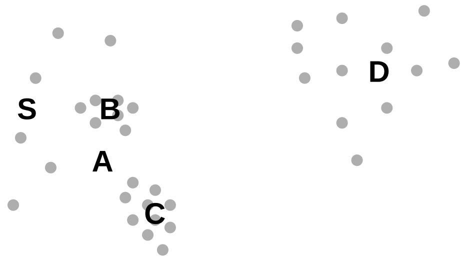


Fig. 5. The high-level cluster A consists of subclusters B and C and a set of separate samples S. The subclusters B and C represent focus on local details, but in larger context they together with S form the high-level cluster A.

To select the best clustering among the partitionings with different number of clusters, the Davies-Bouldin validity index (Eq. 2) was used. In practice, though, it is better to use the index values as a guideline rather than absolute truth. As seen in Fig. 6, the index plots have several local minima. Each sharp local minimum in the validity index plot is certainly worth a look, since it indicates that the addition of one cluster enabled the algorithm to partition the data much better.

D. SOM of data set I

Visual inspection. In Fig. 7, two visualizations of the SOM of data set I are shown: the U-matrix and the map units in the input space. Corresponding clusters in both figures have been encircled and labeled. From the U-matrix, one can easily detect seven distinct clusters, two of which ('O1' and 'O2') are very small and correspond to outliers. The interpolating units are positioned neatly between the clusters, although only the two outliers are completely separated from the rest of the map. There are also some interpolating units within the cluster corresponding to right eye ('E'), possibly indicating presence of two or more subclusters. Note that the relative size of the clusters is proportional to the number of samples in each cluster — not to the radius of the cluster in the input space.

Agglomerative clustering. The full and pruned dendrograms of the SOM of data set I are shown in Fig. 8. The procedure produced 15 interesting clusters, whose hierarchy is shown in Fig. 9(a). The final partitioning was based on the pruned dendrogram: the results, excluding the interpolating units, are shown in Figs. 9(b) and (c).

The result is satisfactory. In the final partitioning, all the original clusters have been found, even the three subclusters of the right eye. However, there is one additional cluster in the right eye, so the performance is not perfect. The right eye causes problems in the SOM-based approach because it is in an extreme corner of the data distribution and the amount of data in that part of the input space is relatively small. If better accuracy is desired, a new clustering can be made using data in that area only. Also note that if the neighborhood constraint had been used in clustering, the outlying map units would have formed two small clusters of their own because they would have been completely separated from the rest of the map.

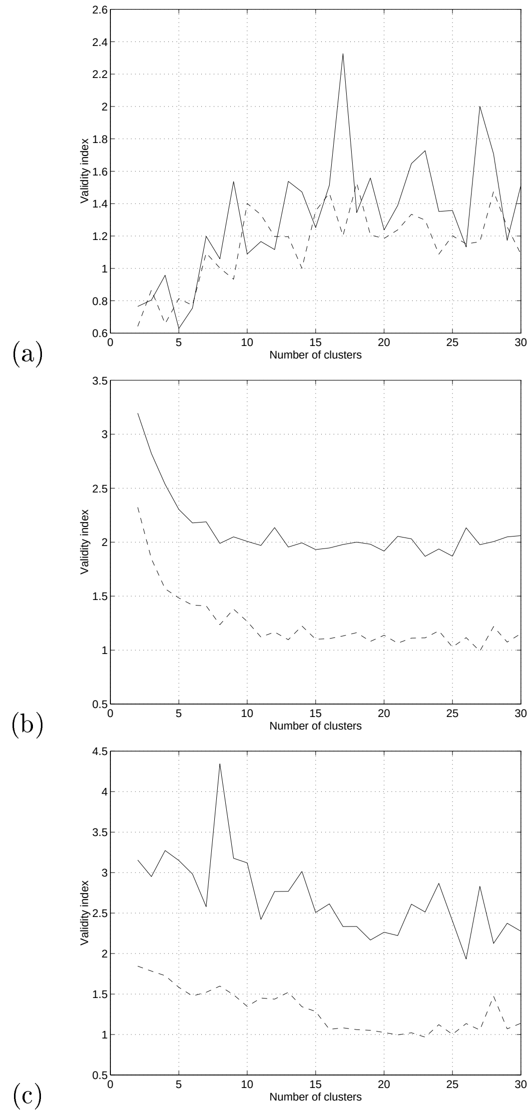
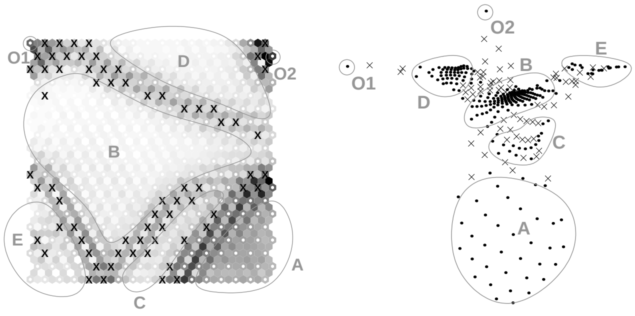


Fig. 6. Davies-Bouldin index for partitive clustering of data sets I–III (a)–(c) as function of the number of clusters. In each plot, the number of clusters runs on horizontal axis and Davies-Bouldin index values lie on vertical axis. Clustering results for the data are marked by solid line and results for the prototype vectors of the SOM by dashed line, respectively.

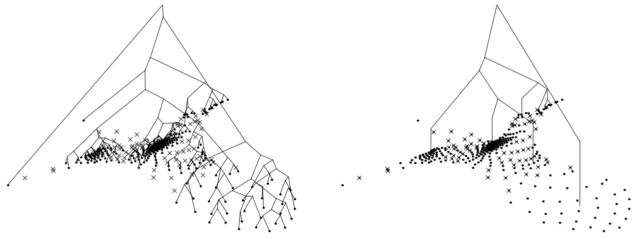
Partitive clustering. According to the Davies-Bouldin index in Fig. 6(a), the number of clusters is small, say, below ten. The index computed from the clustering result of the data has a negative peak at five clusters, whereas the corresponding index of the prototypes of the SOM has similar peak at four clusters. Actually at all clusters below 15 there is one cluster shift in the results. The reason for this off-by-one observation is that the direct clustering groups the outliers on the left to one cluster, while the two-level approach joins them to the left eye cluster. The clustering results using five clusters for the whole data and four clusters for SOM are presented in Fig. 10.

Both clustering the data directly and clustering of prototype vectors using k-means gave reasonable results, but the following problems were observed.



(a) (b)

Fig. 7. The U-matrix (a) and the prototype vectors of the SOM of data set I in input space (b). In the U-matrix, dark color indicates large distance between adjacent map units and light color small distance, respectively. Data clusters have been encircled in both figures and the corresponding ones have been given the same labels. The interpolating map units have been marked with 'X's. Note that the interpolating map units completely separate the two outliers ('O1' and 'O2') from the rest of the map.



(a) (b)

Fig. 8. The full dendrogram (a) and pruned dendrogram (b) of the SOM of data set I. The pruning is essential because it reduces the number of different clusterings into a reasonable level.

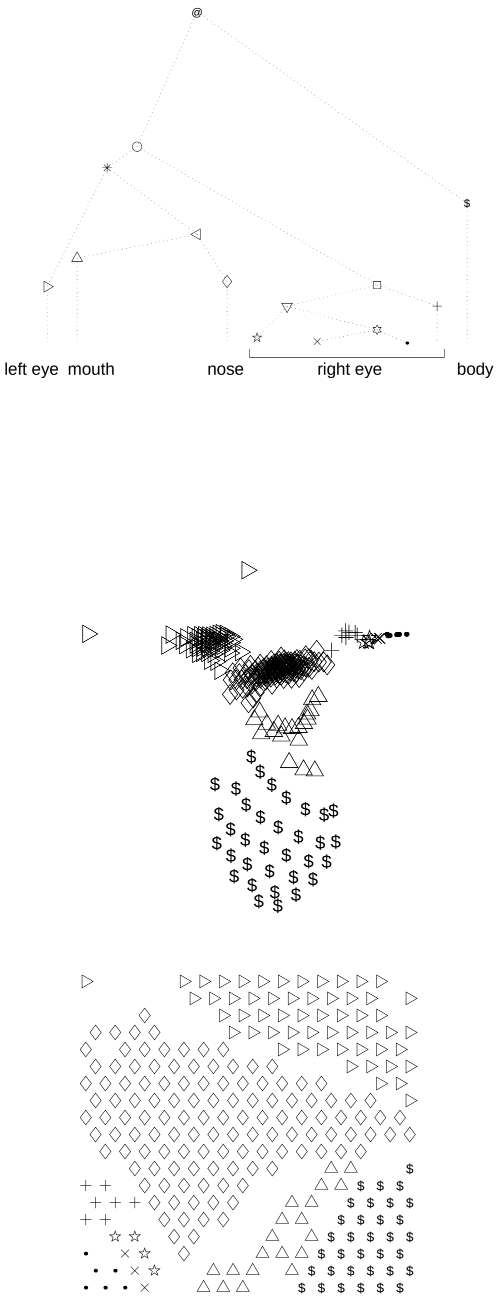
- The three small clusters close to each other in the right eye were not discovered as separate clusters but as a single cluster: as a matter of fact, they were not separated until over 20 clusters were used (not shown).
- The non-spherical cluster could not be properly recognized as one cluster.

Both difficulties originate from properties of the k-means algorithm: it not only seeks for spherical but also clusters with roughly equal number of samples. Therefore, the k-means should never be blindly used to cluster any data without careful verification of the results. It should also be noted that as the number of clusters is increased, the number of samples in clusters decreases, which makes algorithm more sensitive to outliers.

E. SOM of data set II

U-matrix and Sammon's mapping of the SOM of data set II are presented in Figs. 11(a) and (b). According to them there are no clusters: neither visualization reveals any cluster structure, and there are only four interpolating units randomly positioned around the map.

In Fig. 11(c), the partitioning obtained by the agglomer-



(a) (b) (c)

Fig. 9. Results of agglomerative clustering of the SOM of data set I. The interpolating units have been left out of the analysis. The pruned dendrogram (a) has 15 different clusters: each node and leaf represents one "interesting" cluster. The same symbols have been used to indicate cluster memberships of map units in the input space (b) and on the map grid (c).

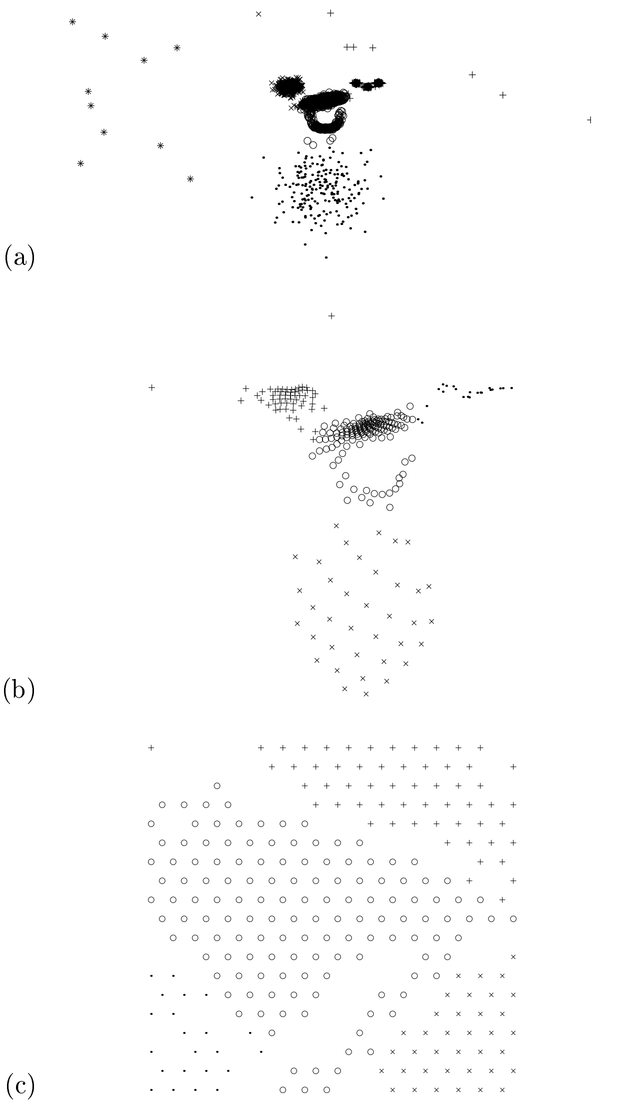


Fig. 10. Results of partitive clustering of the data set I (a) as well as the corresponding SOM in the input space (b) and on the map grid (c).

ative clustering is shown. The result is perfect: except for interpolative units which were left out of the analysis, all units belong to the same cluster. In the partitive clustering, the Davies-Bouldin index reduced to almost constant level as the number of cluster centers increased, see Fig. 6(b). Even though this does not indicate presence of only one cluster, it suggests that the data is somehow suspicious. Clustering the SOM of data set II using four clusters is shown in Fig. 11(d). As might be expected, the clustering nicely splits the SOM into four parts which are almost of equal size.

F. SOM of data set III

U-matrix and Sammon's mapping of the SOM of data set III are presented in Fig. 12. From the U-matrix one can

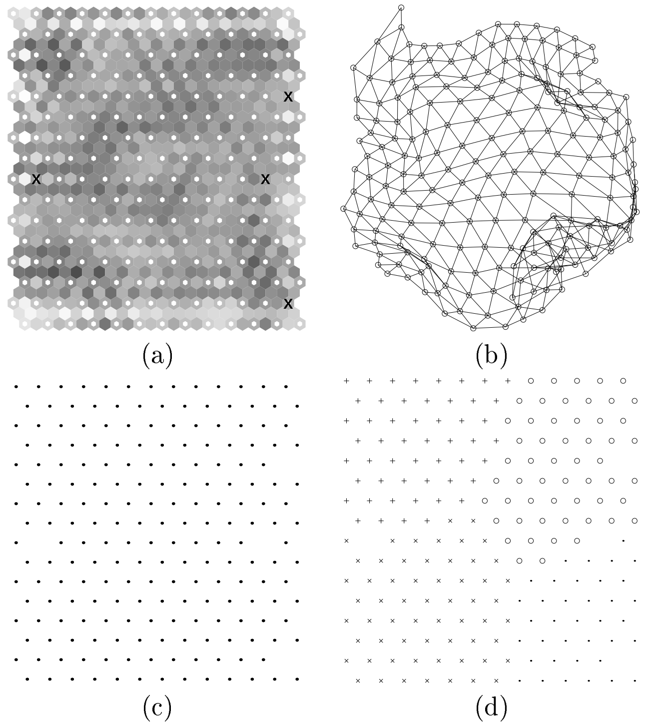


Fig. 11. Clustering of the SOM of data set II: U-matrix (a), Sammon's mapping (b), partitioning using agglomerative clustering (c) and partitioning using partitive clustering with 4 clusters (d). In the Sammon's mapping, the map units have been indicated by circles and neighboring map units have been connected to each other with lines.

clearly distinguish several — maybe about 20 — separate areas. The interpolating map units (marked by 'X's on the U-matrix) are mostly positioned on the borders of the U-matrix. Sammon's mapping is badly folded, but also there one can see several concentrations of points. The folding itself indicates that the actual dimension of the data set is much higher than two, the output dimension used in the projection.

In Figs. 13(a) and (b), the results of agglomerative clustering of SOM of data set III using average and single linkage are shown. The partitionings agree relatively well with the U-matrix in Figure 12(a).

The shape of Davies-Bouldin index plot of data set III in Fig. 6(c) is somewhat similar to the one of data set II in Fig. 6(b). However, there are negative peaks at 6, 10, and 23 clusters. In Figs. 13(c) and (d), the partitioning of the SOM to 10 and 23 clusters is shown. It seems that compared to the U-matrix in Fig. 12, both give reasonable interpretations of the SOM at two accuracy levels.

G. Comparison with direct clustering

Computational load. To estimate the computational load, the execution times were measured. The tests were run in Matlab 5 environment in a SGI O2000 computer using one R10000 processor and enough memory to avoid swapping. While the times do not necessarily represent the true times when using optimized compiled code, they give

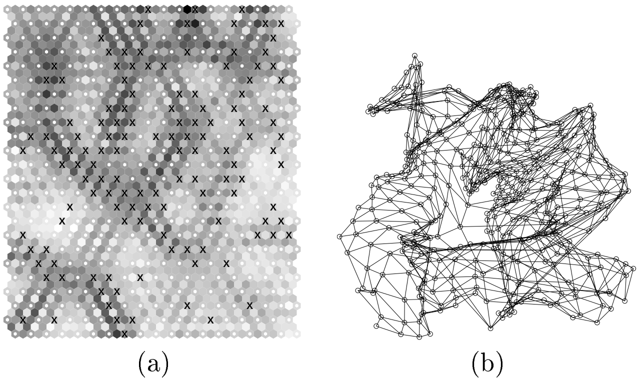


Fig. 12. U-matrix (a) and Sammon's mapping (b) of the SOM of data set III.

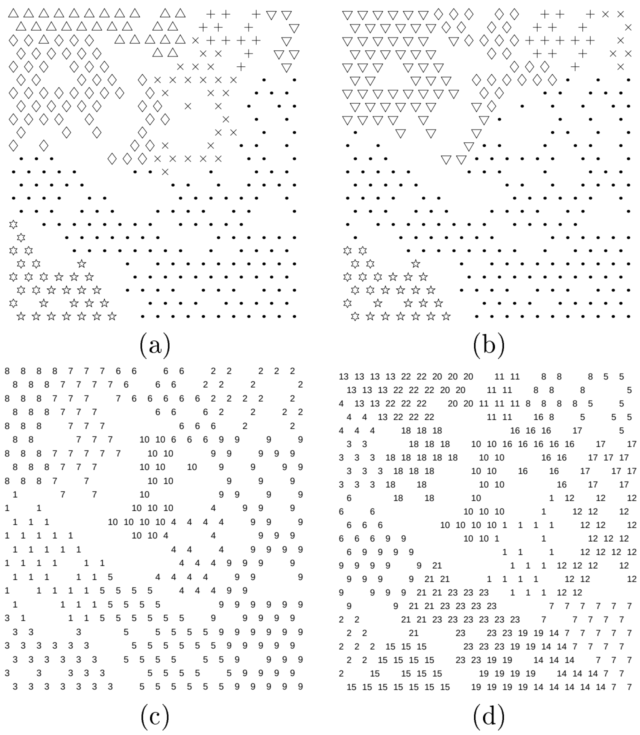


Fig. 13. Partitioning of the SOM of data set III: with agglomerative clustering using average linkage (a) and single linkage (b), and with partitive clustering using 10 (c) and 23 clusters (d).

a fair indication of the relative load of different algorithms. The computation times for SOMs and for dendrogram construction and k-means partitioning to 2–30 clusters (each using 100 runs with different initializations) are listed in Table III. The decrease in computation time is up to two orders of magnitude for agglomerative clustering using average linkage (e.g., 13 hours vs. 10 minutes for data set III). For single and complete linkage, the decrease is smaller, but still remarkable, especially for larger data sets (e.g., 27 vs. 1 minute for data set I). Exact comparison with partitive clustering is more difficult since the number of k-means runs plays a major role. Furthermore, the sequential SOM algorithm was used as opposed to batch k-means. If batch SOM algorithm had been used, the SOM training times would have been much lower. However, for the testing procedure, the saving was about one order of magnitude (e.g., 84 vs. 7 minutes for data set I).

TABLE III

EXECUTION TIMES OF DIFFERENT PARTITIONINGS. THE TIMES FOR k-MEANS ARE FOR 29 PARTITIONINGS $k = 2, \ldots, 30$ USING 100 RUNS WITH DIFFERENT INITIALIZATIONS. THE COMPUTATION TIMES FOR CONSTRUCTING THE SOMS ARE SHOWN IN THE FIRST COLUMN.

data set	single linkage	complete linkage	average linkage	k-means (100 runs)
I	27 min	27 min	1.9 h	84 min
II	2.3 min	2.3 min	9.4 min	32 min
III	3.2 h	3.1 h	13 h	9.0 h
SOM of I (45 s)	1.9 s	1.9 s	5.5 s	6.7 min
SOM of II (20 s)	1.2 s	1.2 s	3.1 s	3.8 min
SOM of III (9.5 min)	11 s	11 s	19 s	10 min

Clustering accuracy. The clustering results were compared using relative conditional entropy. Entropy quantifies the amount of uncertainty present in the state of a variable or, in this application, uncertainty of the cluster of the samples. Entropy of a clustering $\mathbf{Q}$ is defined as $H(\mathbf{Q}) = -\sum_{i=1}^{C} p_i \log(p_i)$, where p_i is the probability of cluster i and C is the number of clusters in $\mathbf{Q}$. The entropy gets its maximum value $\log(C)$ when each cluster is equally probable. Relative entropy is defined $H_r = H/\log(C)$ and has values between $[0, 1]$. As an example, a binary variable with probabilities $p_1 = \{0.5, 0.1, 0.02, 0.01\}$ has relative entropies $H_r = \{1, 0.47, 0.14, 0.08\}$, respectively. The entropy measure used in the tests was relative conditional entropy $H_r(\mathbf{X}|\mathbf{Y})$, which describes uncertainty of clustering $\mathbf{X}$ if clustering $\mathbf{Y}$ is known:

$$H_r(\mathbf{X}|\mathbf{Y}) = -\frac{1}{log(C_X)} \sum_{j=1}^{C_Y} p_j \sum_{i=1}^{C_X} p_{i|j} \log(p_{i|j}), \quad (7)$$

where i and j denote clusters and C_X and C_Y the total number of clusters in $\mathbf{X}$ and $\mathbf{Y}$, respectively.

In Table IV the direct and SOM-based clustering results of data set I are compared with the true known partitioning and with each other. In general the match is very good, although the results are slightly worse for single and complete linkage. Fig. 14 shows the comparison results for partitive clustering of each data set and for agglomerative clustering of data set III. In all cases knowledge of the SOM clustering result has significantly reduced the uncertainty. For data sets I and III, the relative conditional entropies indicate a good correspondence between the two clustering results, but for data set II the results are poor. However, there is a natural explanation. Because data set II actually had no clusters, there is no reason why cluster centers from different clustering runs would match each other, especially with low number of clusters.

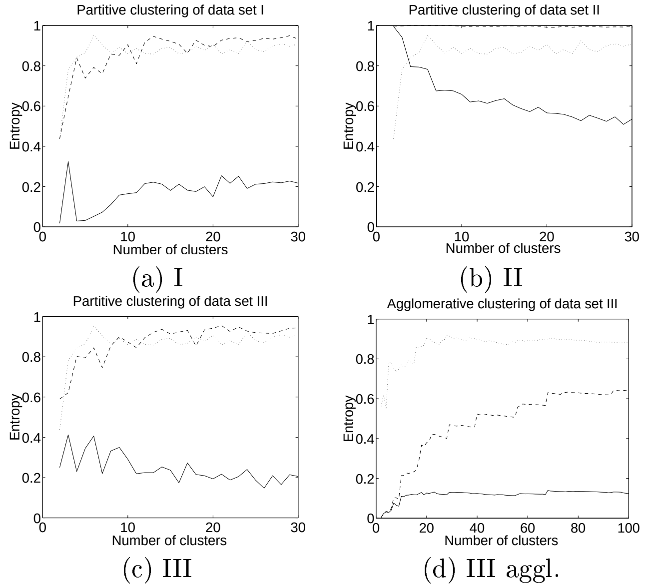


Fig. 14. Results for partitive clustering of data set I (a), II (b) and III (c), and agglomerative clustering of data set III (d). Dashed line is $H_r(\mathbf{Y})$, dotted line $H_r(\mathbf{Z})$ and solid line $H_r(\mathbf{Y}|\mathbf{Z})$, where $\mathbf{Y}$ and $\mathbf{Z}$ are direct and SOM-based partitionings, respectively, with equal number of clusters. The agglomerative clustering in (d) was obtained using complete linkage. Results for single and average linkage were similar. Partitioning was selected by simply cutting the dendrogram at a certain level. From (d), the degrading effect of outliers on agglomerative clustering can be seen: the relative entropy $H_r(\mathbf{Y})$ is very low when the number of clusters is small. This is because vast majority of the samples belongs to a single cluster and only a few samples — the outliers — belong to the others. Partitioning $\mathbf{Z}$ based on the SOM is much more evenly distributed.

In the light of these results, correspondence between direct and two-level clustering can be considered acceptable.

TABLE IV

Relative conditional entropies for data set I. $\mathbf{X}$ is the true partitioning of the data (outliers have been ignored), $\mathbf{Y}$ is the direct partitioning of the data and $\mathbf{Z}$ is the partitioning obtained by clustering the SOM.

clustering	$H_r(\mathbf{X}\|\mathbf{Y})$	$H_r(\mathbf{X}\|\mathbf{Z})$	$H_r(\mathbf{Y}\|\mathbf{Z})$
average linkage	0.020	0.032	0.018
single linkage	0.024	0.031	0.34
complete linkage	0.0098	0.14	0.28
k-means ($k = 4$)	0.22	0.22	0.029
k-means ($k = 7$)	0.12	0.12	0.073

V. Conclusions

In our earlier work it has become obvious that the SOM is an effective platform for visualization of high-dimensional data. However, to be able to fully understand contents of a data set it is vital to find out if the data has cluster structure. If this is the case, the clusters need to be extracted to be able to fully exploit the properties of the data set by producing summary information. For each cluster, not only means (or medians) and ranges of single variables are interesting but also aspects like:

- Which variable(s) make the cluster different from the neighboring clusters?
- Which factors make the cluster different from the rest of the data?
- What is effective data dimension of the cluster?
- Is the cluster spherical, an outlier, and does it have subclusters?
- What are the dependencies between variables in the clusters?

The purpose of this article was to evaluate if the data abstraction created by the SOM could be utilized in clustering of data. The SOM as the first abstraction level in the clustering has some clear advantages. First, the original data set is represented using a smaller set of prototype vectors, which allows efficient use of clustering algorithms to divide the prototypes into groups. The reduction of the computational cost is especially important for hierarchical algorithms allowing clusters of arbitrary size and shape. Second, the two-dimensional grid allows rough visual presentation and interpretation of the clusters.

In the experiments, agglomerative and partitive (k-means) clustering algorithms were run both directly for data and for SOM trained using the data using three data sets. The experiments indicated that clustering the SOM instead of directly clustering the data is computationally effective approach. The clustering results using SOM as an intermediate step were also comparable with the results obtained directly from the data.

References

[1] Dorian Pyle, *Data Preparation for Data Mining*, Morgan Kaufmann Publishers, 1999.

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

[3] Juha Vesanto, "SOM-Based Data Visualization Methods," *Intelligent Data Analysis*, vol. 3, no. 2, pp. 111–126, 1999.

[4] James C. Bezdek and Sankar K. Pal, Eds., *Fuzzy Models for Pattern Recognition: Methods That Search for Structures in Data*, IEEE Press, 1992.

[5] Geoffrey J. McLahlan and Kaye E. Basford, *Mixture Models: Inference and Applications to Clustering*, vol. 84 of *Statistics: Textbooks and Monographs*, Marcel Dekker, 1987.

[6] James C. Bezdek, "Some New Indexes of Cluster Validity," *IEEE Trans. on Systems, Man, and Cybernetics—Part B: Cybernetics*, vol. 28, no. 3, pp. 301–315, 1998.

[7] Marcelo Blatt, Shai Wiseman, and Eytan Domany, "Superparamagnetic Clustering of Data," *Physical Review Letters*, vol. 76, no. 18, pp. 3251–3254, 1996.

[8] George Karypis, Eui-Hong (Sam) Han, and Vipin Kumar, "Chameleon: Hierarchical Clustering Using Dynamic Modeling," *IEEE Computer*, vol. 32, no. 8, pp. 68–74, Aug. 1999.

[9] J. Lampinen and E. Oja, "Clustering Properties of Hierarchical Self-Organizing Maps," *Journal of Mathematical Imaging and Vision*, vol. 2, no. 2-3, pp. 261–272, November 1992.

[10] Eric Boudaillier and Georges Hebrail, "Interactive Interpretation of Hierarchical Clustering," *Intelligent Data Analysis*, vol. 2, no. 3, 1998, Electronic journal at `http://www.elsevier.com/locate/ida`.

[11] Joachim Buhmann, "Complexity Optimized Data Clustering by Competitive Neural Networks," *Neural Computation*, vol. 5, no. 3, pp. 75–88, May 1993.

[12] Glenn W. Milligan and Martha C. Cooper, "An Examination of Procedures for Determining the Number of Clusters in a Data Set," *Psychometrika*, vol. 50, no. 2, pp. 159–179, June 1985.

[13] David L. Davies and Donald W. Bouldin, "A Cluster Separation Measure," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979.

[14] Stephen P. Luttrell, "Hierarchical Self-Organizing Networks," in *Proc. 1st IEE Conf. of Artificial Neural Networks*, London, UK, 1989, pp. 2–6, British Neural Network Society.
[15] A. Varfis and C. Versino, "Clustering of Socio-Economic Data with Kohonen Maps," *Neural Network World*, vol. 2, no. 6, pp. 813–834, 1992.
[16] Paul Mangiameli, Shaw K. Chen, and David West, "A comparison of SOM Neural Network and hierarchical clustering methods," *European Journal of Operational Research*, vol. 93, no. 2, Sept. 1996.
[17] Teuvo Kohonen, "Self-Organizing Maps: Optimization Approaches," In Kohonen et al. [50], pp. 981–990.
[18] John Moody and Christian J. Darken, "Fast Learning in Networks of Locally-Tuned Processing Units," *Neural Computation*, vol. 1, no. 2, pp. 281–294, 1989.
[19] Jari A. Kangas, Teuvo K. Kohonen, and Jorma T. Laaksonen, "Variants of Self-Organizing Maps," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 93–99, Mar. 1990.
[20] Thomas Martinez and Klaus Schulten, "A "Neural-Gas" Network Learns Topologies," In Kohonen et al. [50], pp. 397–402.
[21] Bernd Fritzke, "Let it grow – self-organizing feature maps with problem dependent cell structure," In Kohonen et al. [50], pp. 403–408.
[22] Yizong Cheng, "Clustering with Competing Self-Organizing Maps," in *Proceedings of International Conference on Neural Networks*, 1992, vol. 4, pp. 785–790.
[23] Bernd Fritzke, "Growing Grid – a self-organizing network with constant neighborhood range and adaptation strength," *Neural Processing Letters*, vol. 2, no. 5, pp. 9–13, 1995.
[24] Justine Blackmore and Risto Miikkulainen, "Visualizing High-Dimensional Structure with the Incremental Grid Growing Neural Network," in *Machine Learning: Proceedings of the 12th International Conference*, A. Prieditis and S. Russell, Eds. 1995, pp. 55–63, Kaufmann.
[25] Stefan Jockusch, "A neural network which adapts its stucture to a given set of patterns," in *Parallel Processing in Neural Systems and Computers*, R. Eckmiller, G. Hartmann, and G. Hauske, Eds., pp. 169–172. Elsevier Science Publishers, 1990.
[26] Joaquim S. Rodrigues and Luis B. Almeida, "Improving the learning speed in topological maps of pattern," in *Proceeding of International Neural Network Conference*, 1990, pp. 813–816.
[27] D. Alahakoon and S. K. Halgamuge, "Knowledge Discovery with Supervised and Unsupervised Self Evolving Neural Networks," in *Proceedings of the 5th International Conference on Soft Computing and Information/Intelligent Systems*, T. Yamakawa and G. Matsumoto, Eds. 1998, pp. 907–910, World Scientific.
[28] Allen Gersho, "Asymptotically Optimal Block Quantization," *IEEE Transactions on Information Theory*, vol. IT-25, no. 4, pp. 373–380, July 1979.
[29] Paul L. Zador, "Asymptotic Quantization Error of Continuous Signals and the Quantization Dimension," *IEEE Transactions on Information Theory*, vol. IT-28, no. 2, pp. 139–149, Mar. 1982.
[30] Helge Ritter, "Asymptotic Level Density for a Class of Vector Quantization Processes," *IEEE Transactions on Neural Networks*, vol. 2, no. 1, pp. 173–175, Jan. 1991.
[31] Teuvo Kohonen, "Comparison of SOM Point Densities Based on Different Criteria," *Neural Computation*, vol. 11, pp. 2171–2185, 1999.
[32] R. D. Lawrence, G. S. Almasi, and H. E. Rushmeier, "A Scalable Parallel Algorithm for Self-Organizing Maps with Applications to Sparse Data Problems," *Data mining and knowledge discovery*, vol. 3, no. 2, pp. 171–195, June 1999.
[33] Teuvo Kohonen, Samuel Kaski, Krista Lagus, Jarkko Salojärvi, Jukka Honkela, Vesa Paatero, and Antti Saarela, "Self organization of a massive document collection," *IEEE Transactions on Neural Networks*, 2000, this issue.
[34] Pasi Koikkalainen, "Fast deterministic self-organizing maps," in *Proceedings of International Conference on Artificial Neural Networks*, 1995, vol. 2, pp. 63–68.
[35] A. Ultsch and H. P. Siemon, "Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis," in *Proc. INNC'90, Int. Neural Network Conf.*, Dordrecht, Netherlands, 1990, pp. 305–308, Kluwer.
[36] Martin A. Kraaijveld, Jianchang Mao, and Anil K. Jain, "A Nonlinear Projection Method Based on Kohonen's Topology Preserving Maps," *IEEE Trans. on Neural Networks*, vol. 6, no. 3, pp. 548–559, 1995.
[37] Jukka Iivarinen, Teuvo Kohonen, Jari Kangas, and Sami Kaski, "Visualizing the Clusters on the Self-Organizing Map," in *Proc. Conf. on Artificial Intelligence Res. in Finland*, Christer Carlsson, Timo Järvi, and Tapio Reponen, Eds., Helsinki, Finland, 1994, number 12 in Conf. Proc. of Finnish Artificial Intelligence Society, pp. 122–126, Finnish Artificial Intelligence Society.
[38] Xuegong Zhang and Yanda Li, "Self-Organizing Map as a New Method for Clustering and Data Analysis," in *Proceedings of 1993 International Joint Conference on Neural Networks*, 1993, pp. 2448–2451.
[39] J. B. Kruskal, "Multidimensional Scaling by Optimizing Goodness of Fit to a Nonmetric Hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, Mar. 1964.
[40] J. B. Kruskal, "Nonmetric Multidimensional Scaling: a Numerical Method," *Psychometrika*, vol. 29, no. 2, pp. 115–129, June 1964.
[41] John W. Sammon, Jr., "A nonlinear mapping for data structure analysis," *IEEE Trans. on Computers*, vol. C-18, no. 5, pp. 401–409, May 1969.
[42] P. Demartines and J. Hérault, "Curvilinear Component Analysis: A Self-Organizing Neural Network for Nonlinear Mapping of Data Sets," *IEEE Trans. on Neural Networks*, vol. 8, no. 1, pp. 148–154, January 1997.
[43] Aristide Varfis, "On the use of two traditional statistical techniques to improve the readibility of Kohonen Maps," in *Proc. of NATO ASI workshop on Statistics and Neural Networks*, 1993.
[44] Samuel Kaski, Jarkko Venna, and Teuvo Kohonen, "Coloring that Reveals High-Dimensional Structures in Data," in *Proc. of ICONIP'99, 6th International Conference on Neural Information Processing*, pp. 729–734, 1999.
[45] F. Murtagh, "Interpreting the Kohonen self-organizing map using contiguity-constrained clustering," *Pattern Recognition Letters*, vol. 16, pp. 399–408, 1995.
[46] Olli Simula, Juha Vesanto, Petri Vasara, and Riina-Riitta Helminen, *Industrial Applications of Neural Networks (L.C. Jain and V.R. Vemuri, eds.)*, chapter 4: The Self-Organizing Map in Industry Analysis, pp. 87–112, CRC Press, 1999.
[47] Jacques Vaisey and Allen Gersho, "Simulated annealing and codebook design," in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 1988, pp. 1176–1179.
[48] J. Kelly Flanagan, Darryl R. Morrell, Richard L. Frost, Christopher J. Read, and Brent E. Nelson, "Vector Quantization Codebook Generation Using Simulated Annealing," in *Proceedings of International Conference on Ccoustics, Speech and Signal Processing*. IEEE, 1989, vol. 3, pp. 1759–1762.
[49] T. Graepel, M. Burger, and K. Obermayer, "Phase Transitions in Stochastic Self-Organizing Maps," *Physical Review E*, vol. 56, pp. 3876–3890, 1997.
[50] T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas, Eds., *Artificial Neural Networks*. Elsevier Science Publishers, 1991.

Juha Vesanto received the M.Sc. (Tech.) degree in technical physics from Helsinki University of Technology, Finland, in 1997. He is currently a researcher and a Ph.D. student at the Neural Networks Research Centre, Helsinki University of Technology, Finland. His research interests focus on data mining, especially visualization and analysis of multivariate data.

Esa Alhoniemi (S'97) received the M.Sc. (Tech.) and Lic.Sc. (Tech.) degrees in computer science and engineering from Helsinki University of Technology, Finland, in 1995 and 1998, respectively. Since 1995, he has been working as researcher in the Neural Networks Research Centre in the Helsinki University of Technology. He is currently preparing his Ph.D. thesis on applications of neural networks in the analysis of industrial processes.