ELSEVIER

Computational Statistics & Data Analysis 38 (2001) 161–180

COMPUTATIONAL STATISTICS & DATA ANALYSIS

www.elsevier.com/locate/csda

Extending the Kohonen self-organizing map networks for clustering analysis

Melody Y. Kiang*

Information Systems Department, College of Business Administration, California State University, 1250 Bellflower Blvd., Long Beach, CA 90840-8506, USA

Received 1 November 2000; received in revised form 1 May 2001

Abstract

The self-organizing map (SOM) network was originally designed for solving problems that involve tasks such as clustering, visualization, and abstraction. While Kohonen's SOM networks have been successfully applied as a classification tool to various problem domains, their potential as a robust substitute for clustering and visualization analysis remains relatively unresearched. We believe the inadequacy of attention in the research and application of using SOM networks as a clustering method is due to its lack of procedures to generate groupings from the SOM output. In this paper, we extend the original Kohonen SOM network to include a contiguity-constrained clustering method to perform clustering based on the output map generated by the network. We compare the result with that of the other clustering tools using a classic problem from the domain of group technology. The result shows that the combination of SOM and the contiguity-constrained clustering method produce clustering results that are comparable with that of the other clustering methods. We further test the applicability of the method with two widely referenced machine-learning cases and compare the results with that of several popular statistical clustering methods.

Keywords: Kohenen SOM networks; Clustering analysis; Artificial intelligence; Decision support systems

1. Introduction

The self-organizing map (SOM) network is a special type of neural network that can learn from complex, multi-dimensional data and transform them into visually

* Tel.: +1-562-985-8944; fax: +1-562-985-4080.
E-mail address: mkiang@csulb.edu (M.Y. Kiang).

0167-9473/01/$ - see front matter
PII: S 0167-9473(01)00040-8

decipherable clusters. The theory of the SOM network is motivated by the observation of the operation of the brain. Various human sensory impressions are neurologically mapped into the brain such that spatial or other relations among stimuli correspond to spatial relations among the neurons organized into a two-dimensional map (Kohonen, 1984, 1989, 1995). The main function of SOM networks is to map the input data from an n-dimensional space to a lower dimensional (usually one or two-dimensional) plot while maintaining the original topological relations. The physical location of points on the map shows the relative similarity between the points in the multi-dimensional space.

Unlike other neural network approaches, the SOM network performs unsupervised training; that is, during the learning process the processing units in the network adjust their weights primarily based on the lateral feedback connections. The more common approach to neural networks required supervised training of the network (i.e., the network is fed with a set of training cases and the generated output is compared with the known correct output). Deviations from the correct output result in adjustment of the processing units' weights. On the other hand, unsupervised learning does not require the knowledge of target values. The nodes in the network converge to form clusters to represent groups of entities with similar properties. The number and composition of clusters can be visually determined based on the output distribution generated by the training process.

While Kohonen's SOM networks have been successfully applied as a classification tool to various problem domains, including speech recognition (Leinonen et al., 1993), image data compression (Manikopoulos, 1993), image or character recognition (Bimbo et al., 1993; Sabourin and Mitiche, 1993), robot control (Walter and Schulten, 1993; Ritter et al., 1989) and medical diagnosis (Vercauteren et al., 1990), its potential as a robust substitute for clustering analysis remains relatively unresearched.

Cluster analysis is a technique for grouping subjects into clusters of similar elements. In cluster analysis, we try to identify similar elements by their attributes. We form groups, or clusters, that are homogeneous and different from other groups. SOM networks combine competitive learning with dimensionality reduction by smoothing the clusters with respect to an a priori grid and provide a powerful tool for data visualization. However, the output of an SOM network does not automatically provide groupings of the points on the map. The current practice is to design the Kohonen SOM map so that the number of nodes on the map matches the desired number of clusters. For example, a 2×2 network has four nodes hence forms four groups. However, often times it is difficult to design a two-dimensional map for a problem with small and/or odd number of clusters (e.g. 3 clusters). When the number of SOM output units is different from the number of clusters expected, additional clustering steps are required to further analyze the output map to derive the appropriate number of groupings. Clustering algorithm of any type can take two approaches: agglomerative or partitive. As mentioned earlier, a common problem when using SOM network is that the number of nodes on the output map is more than the number of target groups. Therefore, in this research, we have chosen the agglomerative approach to recursively merge groups from the Kohonen output until a desired number of clusters is reached.

As the first step in this study, we implemented the original SOM algorithm in C++. This language was selected for its object-oriented approach and its generality to other object-oriented algorithms. The code was verified by independently testing specific components and comparing computer-generated results with hand calculations. Each processing unit in the network and each input pattern were implemented using objects. All calculations are performed through message passing between objects in the program. Small networks were used to verify the overall program, again by comparing computer-generated results with hand calculations. The network training for the experiments was performed on a cluster of IBM RS/6000 mini-computers. Version 7 of the SAS statistical package was used to run the other related statistics.

A contiguity-constrained clustering approach has also been implemented in C++ integrated with the earlier developed SOM program. The method groups the output from SOM to any desired number of clusters based on the user requirement. First, we validate the performance of the contiguity-constrained clustering approach with that of other clustering tools in the context of group technology. Then, we apply the method to two widely tested machine-learning problems, the prediction of the class of Iris plants and the wine recognition data from the Machine Learning Database Repository at the University of California, Irvine. Although the two databases were not designed for clustering tasks and the results derived from any clustering tool (unsupervised learning approach) are very likely to be worse than that of any classification tool (supervised learning approach). However, in order to evaluate and compare the performance of our method with that of other clustering approaches, data sets with known cluster solutions are needed to verify the outcome of each method.

What sets this technique apart from other clustering tools is that SOM offers the flexibility of choosing from multiple grouping alternatives and provides a visual map that allows the decision maker to visually analyze and explain the relationship among the points on the map. The balance of the paper is organized as follows: Section 2 presents the basic concepts of SOM network and illustrates its use as a data-reduction tool. Section 3 describes the contiguity-constrained clustering method we implemented. In Section 4, we test and compare the performance of the combined approach with that of statistical clustering methods using three classic cases from different problem domains. The paper concludes with a summary of our findings.

2. Self-organizing map (SOM) networks

The SOM network typically has two layers of nodes, the input layer and the Kohonen layer. The input layer is fully connected to a two-dimensional Kohonen layer. During the training process, input data are fed to the network through the processing elements (nodes) in the input layer. An input pattern $\boldsymbol{x}_v$ $(v = 1, \ldots, V)$ is denoted by a vector of order m as: $\boldsymbol{x}_v = (x_{v1}, x_{v2}, \ldots, x_{vm})$, where x_{vi} is the ith input signal in the pattern and m is the number of input signals in each pattern. An input pattern is simultaneously incident on the nodes of a two-dimensional Kohonen layer. Associated with the N nodes in the $n \times n$ $(N = n \times n)$ Kohonen layer, is a weight

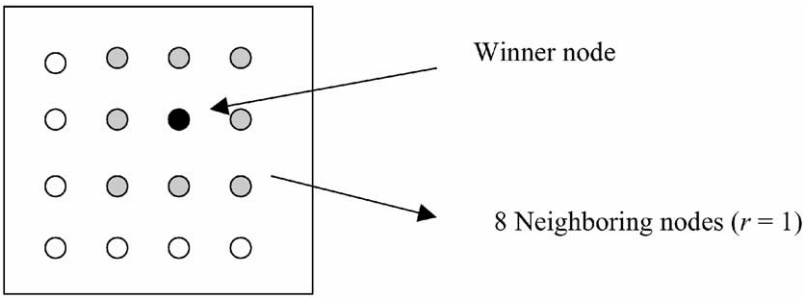


Fig. 1. A 4×4 Kohonen layer and definition of neighboring nodes with the radial distance $(r) = 1$.

vector, also of order m, denoted by: $\boldsymbol{w}_i = (w_{i1}, w_{i2}, \ldots, w_{im})$, where w_{ij} is the weight value associated with node i corresponding to the jth signal of an input vector.

As the training process proceeds, the nodes adjust their weight values according to the topological relations in the input data. The node with the minimum distance is the winner and adjusts its weights to be closer to the value of the input pattern. In this study, Euclidean distance, the most common way of measuring distance between vectors, is used.

2.1. Weight adaptation function

A Gaussian type of neighborhood adaptation function, which decreases both in the spatial domain and the time domain, has been proposed (Cottrell and Fort, 1986; Ritter and Schulten, 1986; Lo and Bavarian, 1991). Lo and Bavarian (1991) have shown that an algorithm that uses the Gaussian type function will enforce ordering in the neighborhood set for every training iteration, yielding faster convergence. They modified Kohonen's adaptation rule to include the amplitude of neighborhood adaptation $A_i(t)$ as follows:

$$\boldsymbol{w}_i(t+1) = \boldsymbol{w}_i(t) + \alpha(t)A_i(t)[\boldsymbol{w}_i(t) - \boldsymbol{x}_v] \quad \text{for } i \in N_i(t),\ 0 <= r <= R,$$
$$\boldsymbol{w}_i(t+1) = \boldsymbol{w}_i(t) \quad \text{otherwise.}$$

Fig. 1 shows the architecture of a 4×4 Kohonen layer and the definition of neighboring nodes with radial distance $(r) = 1$. In general, the farther a node is from the winner, the lower is the amplitude A_i and hence the lower is the update rate of the node's weight vector. For $\alpha(t)A_i(t)$ above, we use a Gaussian type neighborhood adaptation function $h(t, r)$, similar to the one used by Mitra and Pal (1994). This function decreases in both spatial and time domains. In the spatial domain, its value is the largest when node i is the winner node and it gradually decreases with increasing distance from i

$$h(t, r) = \frac{\alpha(1 - rf)}{[1 + (t/cdenom)^2]},$$

where r is the radial distance from the winner node i. Nodes within a radius R are considered for adaptation at time t. Hence, $0 <= r <= R$ if $i \in N_i(t)$. R itself decreases over time and thus fewer and fewer neighbors are updated with every

iteration. Parameter α determines the initial value of $|h|$. The parameter $f(0 < f < 1/r)$ determines the rate of decrease of $|h|$ in the spatial domain. In the time domain, t controls the value of $|h|$ whereas the parameter *cdenom* determines the rate of its decay.

2.2. The self-organization process

The network undergoes a self-organization process through a number of training cycles, starting with randomly chosen $\boldsymbol{w}_i$'s. During each training cycle, every input vector is considered in turn and the winner node is determined. The initial value of R is determined by trial and error and is influenced by the size of the network. Kohonen suggests that the initial size of the neighborhood should be the size of the network itself in order to minimize the effect of initial random weights assigned to the nodes. We followed his recommendation and always set the initial value of R large enough to cover the whole network. The training is conducted in many stages; at each stage, we reduce R by one. Note that R affects the number of nodes in the set N_i. To determine the number of training cycles to be run at each stage, we use the index of disorder D proposed by Mitra and Pal (1994). Essentially, D measures the "improvement" in the "state" of the network at discrete time intervals. The state of the network is denoted by the mean-squared-distance, *msd*, between the input vectors and the weight vectors of the nodes in the set N_i

$$msd = \frac{1}{|trainset|} \sum_{x \in trainset} \left[\sum_{r=0}^{R} \left\{ \left(\frac{1}{|N_r|} \sum_{i \in N_r} ||x - m_i||^2 \right) (1 - rf) \right\} \right],$$

where *trainset* is the training set number and N_r is the number of the set of nodes that are distance r away from the winner node. Note that the nodes closer to the winner node contribute more to *msd*. Due to the converging property of the weight adaptation mechanism, the value of *msd* decreases monotonically with proper choices of α, f, and *cdenom* parameters. *msd* is measured at an interval of every k cycles. The index of disorder D is simply the improvement in *msd* between two consecutive measurements. When this index falls below a certain threshold ($D <$ convergence coefficient δ), the next stage of training begins with a reduced R value. Reader may refer to Mitra and Pal (1994) for the detailed algorithm.

2.3. Conscience mechanism

In a self-organized map, a few nodes may end up representing too much of the input data due to the effect of the initial random weight values assigned to them. To avoid this, we use a "conscience" mechanism that prevents the nodes with higher winning frequency from winning repeatedly and makes the nodes with lower winning frequency more likely to win. The purpose of this mechanism is to give each node in the Kohonen layer an opportunity to represent approximately equal information about the input data.

The conscience mechanism that we use is proposed by DeSieno (1988). It adjusts the Euclidean distance between a node's weight vector and the input vector $||\boldsymbol{x}_v - \boldsymbol{w}_i||$ by a bias B_i. B_i is proportional to the difference between the node's winning frequency and the average winning frequency:

$$B_i = \gamma\left(\frac{1}{N} - F_i\right).$$

F_i is the winning frequency of node i and is updated at every iteration of the training process. Initially, F_i is assigned the average value 1/N; thus $B_i = 0$. The γ coefficient starts at a large value and decreases over time. The winning frequencies are updated as

$$\text{for the winning node}: \quad F_{i,t+1} = F_{i,t} + \beta(1.0 - F_{i,t}),$$

$$\text{for all other nodes}: \quad F_{i,t+1} = F_{i,t} + \beta(0.0 - F_{i,t}),$$

where β is a small positive fraction (see NeuralWare, 1990) NeuralWare Reference Guide, 1990. In this study, we set the value of β fixed at 0.1, a number in the range of appropriate values suggested in NeuralWare (1990).

3. The contiguity-constrained clustering method

When the number of clusters desired is different from the number of nodes on the SOM output map, additional steps are required to analyze and group the points on the output map into the desired number of clusters. Currently, this process is done manually. Sometimes it is hard to visually group the output from SOM especially when the map is highly populated. Hence, a more scientific approach that can help the user to group the output from SOM network based on certain objective criterion is needed. To overcome this limitation, Merkl and Rauber (2000) proposed the Growing-Hierarchical Self-Organizing Map (GH–SOM), a neural network model based on Kohonen SOM network. GH–SOM can grow both in map size and into a three-dimensional tree structure to reflect any hierarchical structure hidden in the underlying data set. The input data are shown in increasingly finer levels of detail along the hierarchy defined by the tree structure. The advantage of their approach is the ability of the output to represent hierarchical structure, if exists, in the data set. However, the added growing capability of the network also increased the computational complexity of the algorithm hence is not as efficient as the original SOM network. In our research, we did not alter the original algorithm of self-organizing process, but extended the model by adding a separate clustering process that takes the output generated by the SOM network to arrive at the desired number of clusters. Moreover, although the tree structure map allows user to visualize if any hierarchical structure exists in the data set, it does not show the relationships among the border nodes that reside in neighboring groups. Therefore, it will be difficult to visually regroup an input to a different cluster or to suggest alternate grouping should it become necessary.

Vesanto and Alhoniemi (2000) also proposed a way to cluster output from SOM. The process starts with a visual inspection of the output map. Then, both agglomerative and partitive clustering algorithms were applied. The neighborhood relations were not used. In the agglomerative clustering, single, average, and complete linkages were used in the construction phase. The partitive clustering was carried out using batch k-means algorithm. In our contiguity-constrained clustering algorithm, it heavily relies on the neighborhood relations when perform clustering. The focus of Vesanto and Alhoniemi's research is on comparing the computational efficiencies of different approaches whereas ours is to introduce the extended SOM network as an alternative clustering tool and compare the performance with those of other popular statistical clustering techniques.

To automate the segmentation process to complement the usage of the Kohonen SOM networks, Murtagh (1995) proposed an agglomerative contiguity-constrained clustering method. The method groups the output from SOM based on a *minimal distance criterion* to merge the neighboring nodes together. The rationale is that the SOM networks will maintain the original topological relations; therefore the nodes that are closely located on the representational grid should have similar cluster centers. In this research, we proposed and implemented a contiguity-constrained grouping algorithm based on a minimal variance criterion that is a better received approach known in traditional statistical clustering methods. The criterion we implemented is modified from Murtagh's (1985) and tries to minimize the overall within cluster variance at each step of the process. We first compare the performance of the system based on the two different grouping criteria. After a few preliminary runs, we found that the minimal variance criterion consistently outperformed the minimal distance approach using our sample cases. Hence, we decided to use the *minimal variance criterion* for our contiguity-constrained clustering method. A comparison of the performance of the two approaches is presented in the Iris plants case study in Section 4.

We start with each node in the map representing one group, and calculate the centroid of each group. Then we try to merge two neighboring groups so the result of the merge will maintain the global minimal variance for that number of clusters. The merge process is repeated until a user specified number of clusters has been derived or when only one cluster remains. The detailed process is described in the following:

Step 1. For each $node_i$, calculate the centroid (C_i) of $node_i$ as

$$C_i = \frac{1}{|node_i|} \sum_{x \in node_i} \vec{x}.$$

where $|node_i|$ is the number of input vectors associated with the node.

Step 2. Assign a group number (G_k) to each $node_i$ if $|node_i| > 0$, and update the corresponding centroid value G_k.

Step 3. Calculate the overall variance of the map:

(a) Sum the square distance between input vector x and the group centroid C_k for all x in G_k. Calculate for every group k

$$V_k = \sum ||x - C_k||^2, \quad x \in G_k.$$

(b) Total the variances from all groups. This will give us the global variance of the map:

$$V_{\text{Total}} = \sum V_k.$$

Step 4. For each pair of neighboring groups, calculate the total variance of the map if the two groups were merged. Merge the two groups that result in the minimum global variance.

(a) Calculate the new centroid for G_{pq} if G_p and G_q were merged:

$$G_{pq} = (|G_p|\vec{G_p} + |G_q|\vec{G_q})/|G_p| + |G_q|.$$

(b) Calculate the new variance if G_p and G_q were merged (modified from Murtagh (1985)):

$$V_{pq} = \sum ||x - C_{\text{new}}|| \quad \text{for all } x,\ x \in G_p \text{ or } x \in G_q.$$

(c) Calculate the new global variance for merging G_p and G_q:

$$V_{pq\text{Total}} = V_{\text{Total}} + V_{pq} - V_p - V_q.$$

(d) Calculate the $V_{pq\text{Total}}$ for every pair of p and q on the map. For each iteration, groups p and q must be within a fixed radial distance on the grid. We start with radial distance = 1, hence for each node there are eight neighboring nodes within that distance (see Fig. 1). We increase the radial distance by one each time if there is no neighboring group within current radial distance for all groups k. Finally, we merge the two groups that result in global minimal variance.

(e) Update V_{Total} and the group number and group centroid of the two newly merged groups.

Step 5. Repeat step 4 until only one cluster or the pre-specified number of clusters has been reached.

4. The comparative study

4.1. The group technology case

We tested our new method using a classic problem in group technology first introduced by Burbidge (1971) in his pioneering work in production flow analysis involving a problem with 43 parts and 16 machines. This case has been studied by many researchers (Chan and Milner, 1982; King, 1980; King and Nakornchai, 1982; Wu et al., 1986; Kulkarni and Kiang, 1995) to compare the quality of clusters formed by their grouping techniques. Burbidge used a hand-computed trial and error method and arrived at a solution with five machine-component clusters. King used his rank order cluster (ROC) algorithm that resulted in a solution with five clusters. An improved version of this approach (ROC2) presented in King and Nakornchai (1982) generated four clusters. Chan and Milner applied their direct clustering algorithm to the same data set to obtain five clusters. Wu et al.'s pattern recognition approach

allowed a limited amount of control over the number of clusters into which the parts can be grouped. Hence, their method generated four different machine-component groupings with four, five, seven, and eight clusters. We use the same case for illustrating the application of the SOM network combined with the contiguity-constrained clustering method to GT and then compare the performance of our technique with that of all of the above methods using the grouping efficacy measure introduced in Kumar and Chandrasekharan (1990).

The data consists of a 16×43 machine-part matrix as shown in Table 1, where each column is labeled with a part and each row is labeled with a machine. A cell entry of 'x' indicates that the concerned part requires an operation performed by the corresponding machine. This matrix is converted into a binary $(0,1)$ matrix and is used directly as input to the SOM network such that each column represents the input vector $x=(x_1,x_2,\ldots,x_m)$ associated with a part. Each value in the vector can be thought of as a coordinate in a 16-dimensional space. The above data set has 43 input vectors corresponding to the 43 parts.

An initial set of experiments is conducted to determine an appropriate configuration for the SOM network. During this stage, various network configurations were tested. The final values of the network parameters set for the SOM network are: network size 7×7, $\alpha = 0.25$, and starting $r = 10.0$ and decreasing over time along with the neighborhood size (R). The other network sizes (9×9, 11×11, and 13×13) we tested gave results similar to the 7×7 network. Similar results were also derived when varying the other parameter values such as α and r. Fig. 2 shows the scatter plot representing the converged SOM for network size 7×7. Each point in the figures represents one or more parts. Fig. 3(a)–(f) show the same output map with the six different clustering designs derived from our contiguity-constrained clustering method. A cluster represents parts grouped together due to their similarity.

We used the grouping efficacy measure, Γ, derived in Kumar and Chandrasekharan (1990), to compare the performance of our technique with the other methods. Grouping efficacy is a measure designed to evaluate the quality of a clustering scheme. From a block-diagonalized machine-part matrix representing a clustering of parts and machines, grouping efficacy, Γ, is computed as

$$\Gamma = \frac{\text{number of } X'\text{s in the diagonal blocks}}{\text{operational zone}},$$

where the operational zone is the area covered by the diagonal blocks plus the number of X's in the off-diagonal region. The higher the value of Γ is, the better the quality of the clustering. Unlike other methods of evaluating the quality of a clustering, grouping efficacy has certain desirable properties, such as (i) non-negativity, (ii) a 0–1 range with a meaningful interpretation of the extreme values, (iii) a built-in relative weight for voids (in the diagonal blocks) and exceptional elements, and (iv) a good discriminating power. For example, a block-diagonalized diagram shown in Table 1 represents a case of 8 machines and 10 parts grouped into 2 families. The number of X's in the diagonal blocks $=28$ and the operational zone $(=45)$ is calculated by adding the area covered by the diagonal blocks $(=40)$ and the number of X's in the off-diagonal region $(=5)$. Therefore, $\Gamma = 28/45 = 62.22\%$. Table 2

Table 1
The part-machine matrix

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
A																																					X				X	
B		X								X																		X				X					X	X		X		X	
C							X										X																	X	X	X							
D					X				X					X					X		X		X					X														
E					X			X	X					X	X	X			X		X		X					X				X							X		X
F	X	X				X	X	X				X	X	X			X		X				X								X	X	X			X		X	X		X	X
G	X												X												X												X						
H	X	X	X					X	X		X	X			X				X	X	X		X	X			X	X		X							X	X		X		X	
I		X		X						X								X									X				X						X	X		X		X	
J	X											X	X												X	X			X											X				
K			X						X											X			X				X			X													
L											X											X	X				X			X													
M			X																					X																			
N		X				X											X																		X								
O					X									X					X		X											X							X		X	
P		X					X			X								X													X					X	X			X		

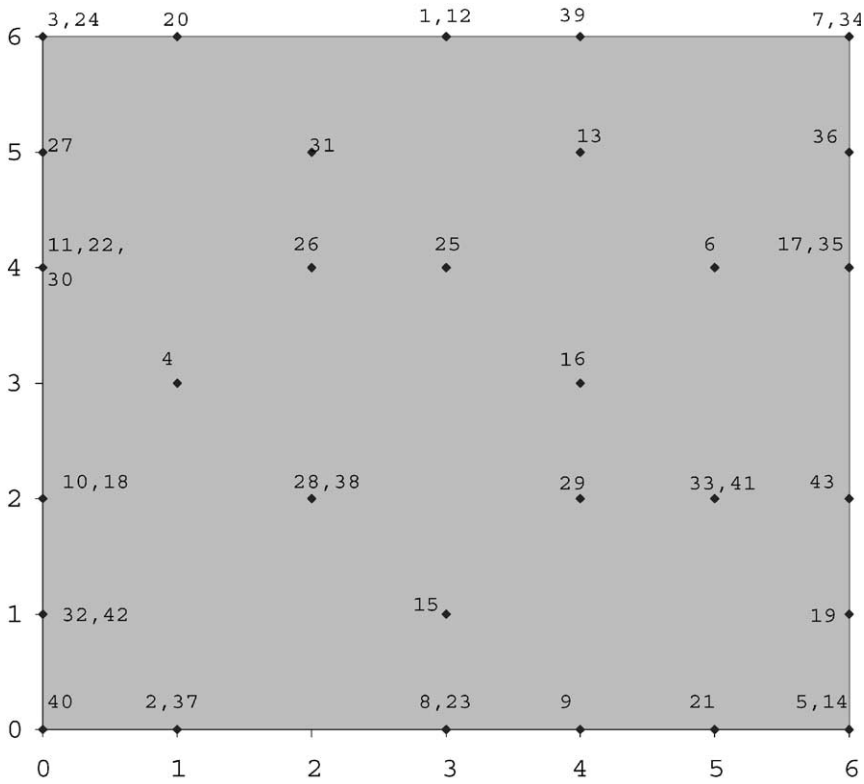


Fig. 2. Output map from SOM network size 7×7.

compares the grouping efficacy, Γ (expressed as a percentage), for the clusterings obtained by the various methods.

Any metric developed to measure the goodness of a clustering design will, in general, find that a clustering design with a large number of clusters (i.e., eight clusters) to be a better design than that with a fewer number of clusters (i.e., three clusters). This is because, for a given data set, a design with a larger number of clusters has fewer parts in each cluster and therefore the parts within a cluster are more likely to be closer to each other. The desired number of clusters is problem dependent. It is therefore fair to compare the performance of different clustering techniques across clustering designs with only an equal number of clusters. For one technique to be judged better than another, it should perform consistently better than the other across different clustering designs.

It can be seen that the grouping efficacy of our method was equal or better than all other methods across three clustering designs (four, five, and eight clusters) (Table 3). For the clustering design with seven clusters, the pattern recognition method performed slightly better than our method, but the difference is insignificant (0.4%). For the four cluster design, our method performed better than the

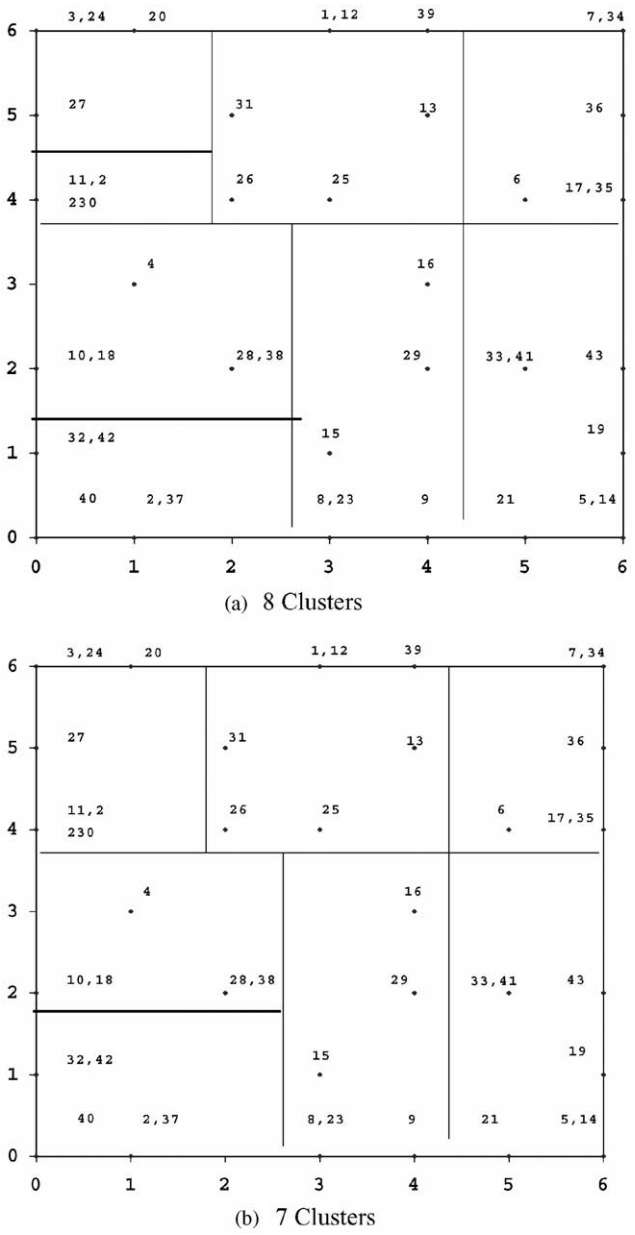
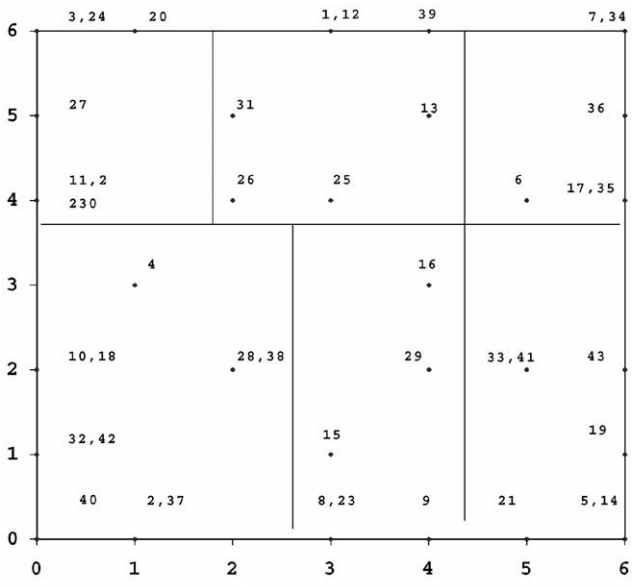
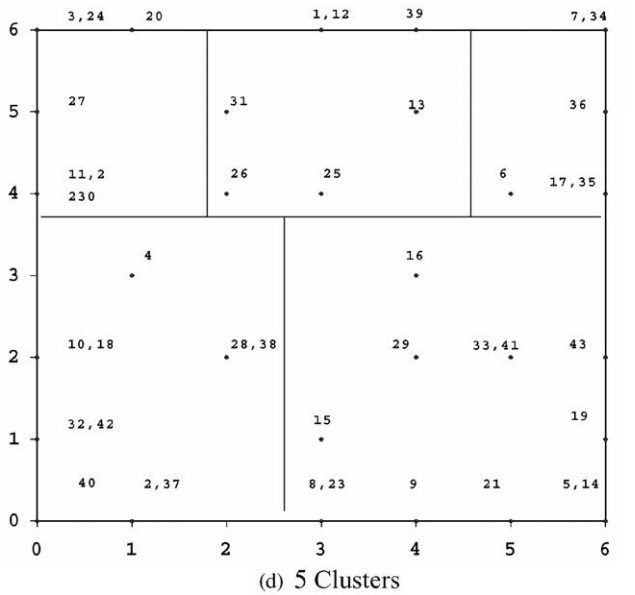


Fig. 3. Six clustering designs for SOM network size 7×7 output Ma.

pattern recognition and ROC2 methods by 3.2% and 2.6%, respectively. Overall, the performance of our method does as well as or better than that of the known best results in most cases. Moreover, the flexibility in the clustering design allows the decision maker to examine the performance of all designs and determine the optimal number of groupings based on both the performance metric and SOM output map distribution.



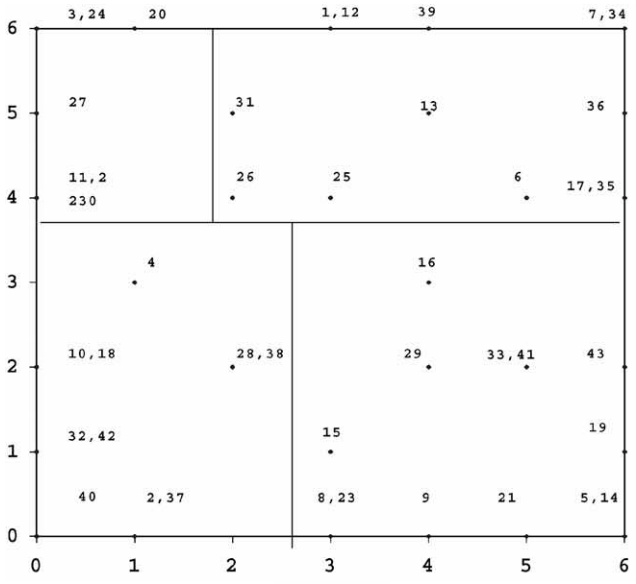
(c) 6 Clusters



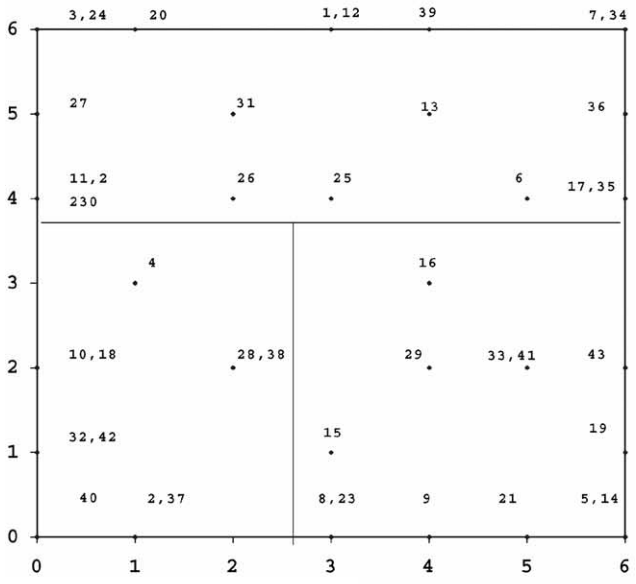
(d) 5 Clusters

Fig. 3. (Continued.)

In the following two cases, we compare the extended SOM network with three statistical clustering methods, both parametric and non-parametric approaches. They are k-means analysis, a popular clustering method based on the least squares criterion, the Ward's minimum variance cluster analysis, and a non-parametric (or distribution-free) method supported by SAS MODECLUS procedure.



(e) 4 Clusters



(f) 3 Clusters

Fig. 3. (Continued.)

4.2. The Iris plants database

The database was created by Fisher (1936) and has been since widely used in subsequent research in pattern classification (Duda and Hart, 1973; Dasarathy, 1980). The data set contains three classes of 50 instances each, where each class refers to a type of Iris plant. The three classes are: *Iris setosa*, *Iris versicolour*, and *Iris*

Table 2
An example part-machine matrix

	1	2	3	4	5	6	7	8	9	10
A	X			X						
B		X	X							X
C	X			X						
D	X		X	X					X	
E					X	X		X	X	X
F			X			X	X	X	X	X
G	X				X	X	X	X	X	X
H		X			X			X	X	

Table 3
Grouping efficacy (%)

	Method					
	PFA[a]	ROC[b]	ROC2	DCA[c]	Pattern recognition	SOM and the contiguity-constrained method
	Source					
Number of clusters	Burbidge	King	King and Nakornchai	Chan and Milner	Wu et al.,	Kiang
3	—	—	—	—	—	39.9
4	—	—	50.0	—	49.4	52.6
5	59.5	60.6	—	60.9	60.9	60.9
6	—	—	—	—	—	63.0
7	—	—	—	—	66.5	66.1
8	—	—	—	—	67.6	67.6

[a]Production flow analysis.
[b]Rank order cluster.
[c]Direct clustering algorithm.

virginica. The first class is linearly separable from the other two; the latter are not linearly separable from each other. There are four numeric attributes and no missing value. The four attributes are: (1) sepal length in cm, (2) sepal width in cm, (3) petal length in cm, and (4) petal width in cm. Table 4 is a brief statistical analysis of the sample attributes.

Based on our previous experiments on the group technology case, the varying of the network parameter values does not have significant effect on the performance of the network. For the rest of the experiments, we decided to fix the network parameter values to: $\alpha = 0.25$ and $r = 10.0$ and which decrease over time. The network sizes of 5×5, 7×7, 9×9, and 11×11 were used and the average of the results is compared with that of the other three statistical approaches.

Table 4
Statistical analysis of the Iris plants sample attributes

Attribute	Min.	Max.	Mean	Standard deviation
Sepal length	4.3	7.9	5.84	0.83
Sepal width	2.0	4.4	3.05	0.43
Petal length	1.0	6.9	3.76	1.76
Petal width	0.1	2.5	1.20	0.76

Table 5
Performance comparison of the extended SOM and the three statistical clustering methods using the Iris database

Method	Rate of correctness (%)
Extended SOM (minimum variance)	90.34
Extended SOM (minimum distance)	89.17
k-means analysis	88.67
Ward's	88.67
MODECLUS	88.67

Table 5 summarizes the experiment results.

We implemented two versions of the extended SOM networks, one uses minimum variance and the other uses minimum distance criterion for clustering. Both versions of the extended SOM networks outperformed the three statistical methods. The minimum variance approach outperformed the minimum distance approach that also concurs with the findings from our preliminary test runs.

The results show that the average performance of the extended SOMS method using minimum variance criterion is significantly better (test of significance, $p < 0.01$) than that of the three statistical methods. Results of the four different networks differ slightly (standard deviation (SD) $= 1.15\%$) which further demonstrates the stability of the performance of our method. Fig. 4 shows the output map of network 9×9.

4.3. The wine recognition data

The data set contains three classes and there are 59, 71, and 48 instances in each class, respectively. These data are the results of a chemical analysis of wines produced in the same region in Italy but derived from three different cultivators. The analysis determined the quantities of 13 constituents (attributes) found in each of the three types of wines. The 13 attributes are: (1) Alcohol, (2) Malic acid, (3) Ash, (4) Alcalinity of ash, (5) Magnesium, (6) Total phenols, (7) Flavanoids, (8) Non-flavanoid phenols, (9) Proanthocyanins, (10) Color intensity, (11) Hue, (12) OD280/OD315 of diluted wines, and (13) Proline. All attributes are continuous and have no missing value. It is also suggested that the attributes be standardized for classifiers that are not scale invariant.

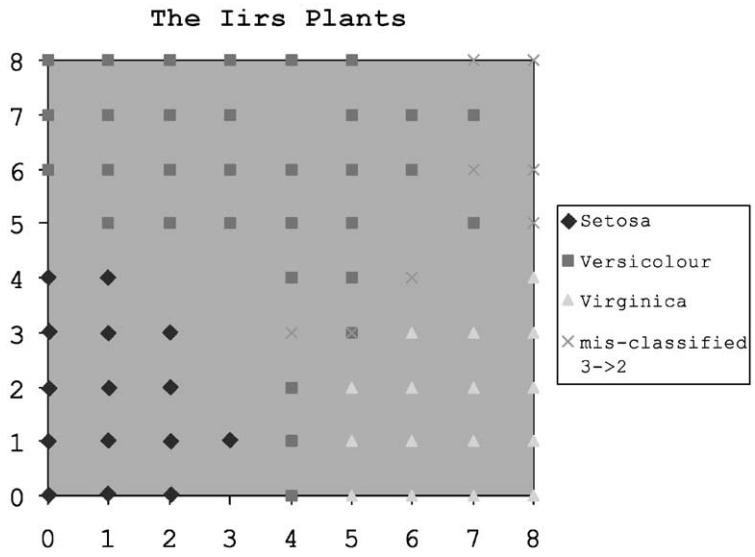


Fig. 4. The output map for the Iris plants data.

Table 6
Performance comparison of the extended SOM with the three statistical clustering methods using the Wine database

Method	Rate of correctness (%)
Extended SOM	93.26
k-means analysis	85.96
Ward's	97.75
MODECLUS	88.67

We followed the suggestion and standardized all the variables to values between 0 and 1 for both methods. To simplify our experiment design, we again fixed the network parameter values to: $\alpha = 0.25$ and $r = 10.0$ and which decrease over time. The same network sizes of 5×5, 7×7, 9×9, and 11×11 were used and the average result is compared with that of the three statistical clustering methods. Table 6 summarizes the experiment results.

The results show that the extended SOM method outperformed k-means analysis and the MODECLUS procedure by 7.3% and 4.59%, respectively. However, the Ward's minimum variance cluster analysis did better than our method by 4.49%. Again, results of the four different networks differ slightly (standard deviation (SD) $= 2\%$). Fig. 5 is the output map of network 5×5.

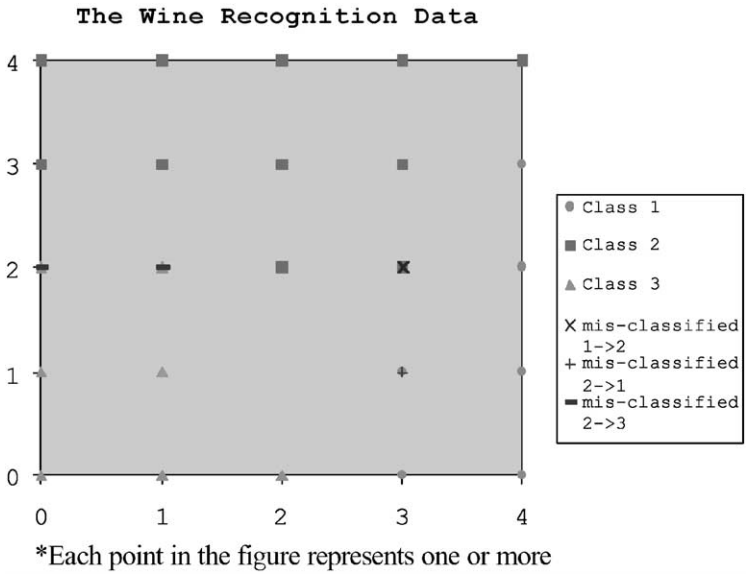


Fig. 5. The output map for the wine recognition data.

5. Conclusion

The advantages of the combined method over other clustering tools are that in addition to the flexibility in determining the number of clusters needed, it also provides a visual map that can be used as a decision-support tool. The map allows the decision makers to visualize the relationship among the subjects being grouped that not only can help in explaining the outputs but also provide the possibility of including expert knowledge in fine-tuning the groupings.

For example, when performing the part family grouping, the output of SOM is a two-dimensional map depicting the relationships between parts such that the parts with similar processing needs are close to each other. This offers an easy way to visualize a picture of the parts to be grouped together into families. In this way, our method cannot only allow control over the number of cells but it also suggests alternative groupings of parts. One of the benefits of the flexibility in grouping is the availability of control over balancing load among cells. Many of the classical clustering techniques do not offer this flexibility. Some of the AI-based techniques do offer flexibility in terms of number of cells and output alternative groupings of some parts, but the parts for which alternative groupings are suggested are not necessarily those that need to be regrouped. Moreover, unlike our method, the output of other techniques is a complete part grouping and does not give the decision maker any insight into the relationships between parts. As a result, it is difficult for the decision maker to make any adjustments to the groupings.

Another advantage of our method is that it can accept both numeric and binary input data. We tested the binary input data using the group technology case and the numeric data using the two machine-learning problems.

In this study, we tested the performance of the combined approach using three cases, a classic case in group technology and two cases from machine learning databases. The preliminary results are encouraging and we believe that by extending the current SOM network with the contiguity-constrained clustering method to group the output from SOM map would make the combined method an appealing and powerful decision-support system tool for clustering tasks.

References

Bimbo, A.D., Landi, L., Santini, S., 1993. Three-dimensional planar-faced object classification with Kohonen maps. Opt. Eng. 32 (6), 1222–1234.

Burbidge, J.L., 1971. Production flow analysis. Prod. Eng. 138–152.

Chan, H.M., Milner, D.A., 1982. Direct clustering algorithm for group formation in cellular manufacturing. J. Manuf. Systems 1/1, 65–75.

Cottrell, M., Fort, J.C., 1986. A stochastic model of retinotopy: a self-organizing process. Biol. Cybern. 53, 405–411.

Dasarathy, B.V., 1980. Nosing around the neighborhood: a new system structure and classification rule for recognition in partially exposed environments. IEEE Trans. Pattern Anal. Mach. Intell. PAMI-2 (1), 67–71.

DeSieno, D., 1988. Adding a conscience to competitive learning. Proceedings of the International Conference on Neural Networks, Vol. I, IEEE Press, New York, 117–124.

Duda, R.O., Hart, P.E., 1973. Pattern Classification and Scene Analysis. Wiley, New York, p. 218.

Fisher, R.A., 1936. The use of multiple measurements in taxonomic problems. Ann. Eugenics 7 (Part II) 179–188.

King, J.R., 1980. Machine-component grouping in production flow analysis: an approach using a rank order clustering algorithm. Int. J. Prod. Res. 18/2, 213–232.

King, J.R., Nakornchai, V., 1982. Maching-component group formation in group technology: review and extension. Int. J. Prod. Res. 20/2, 117–133.

Kulkarni, U.R., Kiang, M.Y., 1995. Dynamic grouping of parts in flexible manufacturing systems—a self-organizing neural networks approach. European J. Oper. Res. (EJOR) 84, 192–212.

Kumar, C.S., Chandrasekharan, M.P., 1990. Grouping efficacy: a quantitative criterion for goodness of block diagonal forms of binary matrices in group technology. Int. J. Prod. Res. 28/2, 233–243.

Kohonen, T., 1984. Cybernetic Systems: Recognition, Learning, Self-Organization. In: Caianiello, E.R., Musso, G. (Eds.), Research Studies Press, Ltd., Letchworth, Herfordshire, UK, p. 3.

Kohonen, T., 1989. Self-Organization and Associative Memory. 2nd Edition. Springer, Berlin.

Kohonen, T., 1995. Self-Organizing Maps. Springer, Berlin.

Leinonen, L., Hiltunen, T., Torkkola, K., Kangas, J., 1993. Self-organized acoustic feature map in detection of fricative-vowel coarticulation. J. Acoust. Soc. Am. 93 (6), 3468–3474.

Lo, Z.-P., Bavarian, B., 1991. On the rate of convergence in topology preserving neural networks. Biol. Cybern. 65, 55–63.

Manikopoulos, C.N., 1993. Finite state vector quantisation with neural network classification of states. IEEE Proc.-F 140 (3), 153–161.

Merkl, D., Rauber, A., 2000. Uncovering the hierarchical structure of text archives by using an unsupervised neural network with adaptive architecture. PADKK, LNAI 1805, pp. 384–395.

Mitra, S., Pal, S.K., 1994. Self-organizing neural network as a fuzzy classifier. IEEE Trans. Systems, Man, Cybernetics 24 (3), 385–399.

Murtagh, F., 1985. Multidimensional Clustering Algorithms. Pysica-Verlag, Wurzburg.

Murtagh, F., 1995. Interpreting the Kohonen self-organizing feature map using contiguity-constrained clustering. Pattern Recognition Lett. 16, 399–408.

NeuralWare Reference Guide, 1990. NeuralWare, Inc., Pittsburgh, PA.

Ritter, H., Schulten, K., 1986. On the stationary state of Kohonen's self-organizing sensory mapping. Biol. Cybern. 54, 99–106.

Ritter, H., Martinetz, T., Schulten, K., 1989. Topology-conserving maps for learning visuo-motor-coordination. Neural Networks 2, 159–168.

Sabourin, M., Mitiche, A., 1993. Modeling and classification of shape using a Kohonen associative memory with selective multiresolution. Neural Networks 6, 275–283.

Vercauteren, L., Sieben, G., Praet, M., Otte, G., Vingerhoeds, R., Boullart, L., Calliauw, L., Roels, H., 1990. The classification of brain tumours by a topological map, Proceedings of the International Neural Networks Conference, Paris, pp. 387–391.

Vesanto, J., Alhoniemi, E., 2000. Clustering of the self-organizing map. IEEE Trans. Neural Networks 11 (3), 586–600.

Walter, J.A., Schulten, K.J., 1993. Implementation of self-organizing neural networks for visuo-motor control of an industrial robot. IEEE Trans. Neural Networks 4 (1), 86–95.

Wu, H.L., Venugopal, R., Barash, M.M., 1986. Design of a cellular manufacturing system: a syntactic pattern recognition approach. J. Manuf. Systems 5/2, 81–88.