A Three-layered Self-Organizing Map Neural Network for Clustering Analysis

Sheng-Chai Chi, Chi-Chung Lee and Tung-Chang Young
Department of Industrial Management
Huafan University
Taipei County, Taiwan, R.O.C.
E-mail: scchi@huafan.hfu.edu.tw

ABSTRCT

In the commercial world today, holding the effective information through information technology (IT) and the internet is a very important indicator of whether an enterprise has competitive advantage in business. Clustering analysis, a technique for data mining or data analysis in databases, has been widely applied in various areas. Its purpose is to segment the individuals in the same population according to their characteristics. In this research, an enhanced three-layered self-organizing map neural network, called 3LSOM, is developed to overcome the drawback of the conventional two-layered SOM through sight-inspection after the mapping process. To further verify its feasibility, the proposed model is applied to two common problems: the identification of four given groups of work-part images and the clustering of a machine/part incidence matrix. The experimental results prove that the data that belong to the same group can be mapped to the same neuron on the output layer of the 3LSOM. Its performance in clustering accuracy is good and is also comparable with that of the FSOM, FCM and k-Means.

Keywords *Neural Network, Self-Organizing Map (SOM), Three-layered SOM, Clustering Analysis, Part Family/Machine Cell Formation*

1. Introduction

Competition among the enterprises depends mainly on manufacturing skills and production costs. Therefore, it is essential to supply products of high quality and low cost through mass production. To achieve this purpose, many have resorted to automation of production, thus improving the quality, and raising the capacity of productivity. In addition, Enterprise Resources Planning (ERP) and Supply Chain Management (SCM) are popularly used to ensure the competitive superiority against the odds.

Clustering analysis, an old and popular technique, has been broadly utilized in various application areas. Its main function is to group those with the same features in the universe into a group such that those in the same group are homogenous and those in different groups are heterogeneous. In this research, we modify the conventional SOM neural network for clustering input data in order to explore useful information or implied knowledge.

2. Literature Review

In the present world, techniques of data mining and clustering analysis have been widely applied for data analysis and knowledge discovery. The purpose is to transform the seemingly irrelevant data in the database into meaningful features and rules by using automatic or semi-automatic approaches. Furthermore, the features and rules can be used for building a sophisticated model or system. The related references are discussed in the following two sub-sections, namely clustering analysis and self-organizing map neural network, SOM.

2.1 Clustering Analysis

Data clustering techniques are very useful and important research methods in data mining. The popular clustering techniques can be classified into three categories: hierarchical clustering algorithms, partitioned (or non-hierarchical) clustering algorithms, and two-phase clustering algorithms. [7]

The common hierarchical clustering algorithms include Single Linkage Method (SLM), Complete Linkage Method (CLM), Average Linkage Method (ALM), Centered Method (CM), Median Method (MM), and Ward Method (WM). Among these methods, the clustering process of SLM, CLM, ALM, CM and MM are similar, but they differ only in the way of calculating the distance. Moreover, the WM has an analysis process similar to that of the SLM, but the combination of any two clusters is dependent on the shortest squared distance between the cluster centers. Using the hierarchical clustering process, Zamir et al. [17] proposed the CLM, which has the best performance of clustering among all the approaches mentioned above although the computation involved is the most complicated.

In the partitioned clustering algorithms, their classification can be discussed in two aspects. One starts from considering the importance of each data point, identifying the most important points, and agglomerating the neighboring points into clusters. This kind of algorithms resembles the possibilitic C-Means (PCM). [10,11] The other considers the importance of the cluster centers and assign the remaining data points into each cluster. This kind of algorithms includes the K-Means and Fuzzy C-Means (FCM). The PCM has one control parameter more than FCM. The purpose is to reduce the influence of noise signals. However, its complexity is relatively higher than that of FCM.

Among these algorithms, the K-Means proposed by MacQueen in 1967 is the most frequently applied. It can be used in multi-dimension, and takes the Euclidean distance as the standard function to calculate the difference among data. It was proven by Selim [15] that the K-Means could only find a local optimal solution such that lots of scholars proposed methodologies of researching the optimal solution to this problem. The approaches include the Simulated Annealing, Genetic Evolution and Evolution Strategy etc. [5,6,9]

Even the above two categories of clustering algorithms have their specific merits, the partitioned algorithms have to pre-determine the number of clusters and their centers (or means); the hierarchical algorithms can hardly decide the correlation of the whole data being allocated into the same cluster. To bridge over the respective defects, a stepwise hybrid method combining the hierarchical and partitioned clustering algorithms was proposed. [16]

For instance, the Genetic Evolution approach was first employed to search the most appropriate number of clusters in the existing data population using the operations of crossover and mutation. Secondly, the conventional K-Means method was used for finding out the global optimal solution of clustering analysis. Therefore, this approach method was named Genetic K-means Algorithm (GKA). [12] In the later research [14], a clustering algorithm combining SOM and GKA was proposed where the cluster amount obtained from grouping by SOM in advance is utilized in genetic evolution for data clustering, and the centers of gravity are rearranged by the K-Means to shorten the time of evolution, sequentially.

In addition, Punj and Steward [13] also suggested a two-stage method for clustering. The feature of a two-stage method is to find out the cluster amounts by taking advantage of the hierarchical clustering algorithm, and then perform clustering with the partitioned clustering algorithm.

2.2 Self-Organizing Map

Self-organizing map neural network is a new technique popularly used for data clustering or data mining. One of its significant features is its capacity of generating automatically an appropriate number of clusters to associate the relations between the original data distribution. Growing Self-Organizing Map (GSOM), a powerful and flexible SOM-related model, was proposed for mining fuzzy rules in recent research [4]. As the result shown, the algorithm of GSOM could get better performance in associating the similar data points than other methods. In Vesanto and Alhoniemi [7], a two-stage approach is also used for data mining. In the first stage, a set of typical prototypes can be generated within the preliminary clustering process and then proceed to the clustering analysis is then performed in the second stage using these prototypes. Additionally, in Kuo et al. [14], the SOM and K-Means were combined into a two-stage method for clustering analysis. The number of clusters and the gravity centers of the clusters are first mapped on the network topology of SOM. After that, the number of clusters obtained is regarded as the pre-set number of groups as well as the gravity centers is considered as the initial centers for the K-Means.

3. Methodology

Due to the drawback that a judgment of cluster amounts and data classification has to be carried out after clustering by means of dotting data points on the topology in the conventional two-layered SOM models, an innovative model by developing a three-layered SOM is proposed in this research, named 3LSOM. The outputs of the neurons on the network topology layer (the second layer) are calculated and transmitted into the third layer (or the output layer) for further mapping by means of formulas to make the clustering results more obvious and precise. The 3LSOM developed in this study is depicted in Figure 1.

In the system network structure, there are two data processing methods used for generating the outputs from the network topology layer in order to further map to the output layer. The first method is to calculate the Euclidean distance between the input vector and the weight vector, which connects one specific neuron on the second layer with the input neurons, and then normalize the distance obtained into a value within the interval [0, 1]. The shorter the distance is, the larger the normalized value is generated. The second method is to calculate the topology coordinate distance between the winning neuron and the other neurons on the two-dimensional network topology layer, and normalize it by the similar formula as well. The term *Stepwise Training Mode* means that the network training process in this approach can be divided into two stages. The first stage is to train the weights between the input layer and the network topology layer before convergence and fix the trained weights. The second stage is to train the weights between the network topology layer and the output layer until convergence.

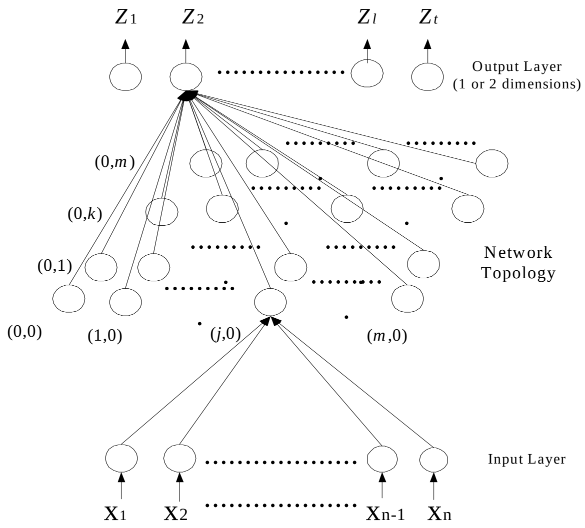

Figure 1. Diagram of the proposed 3LSOM neural network

The system proposed in this research can be divided into two models: Model A and Model B. In the Model A, the first stage is to calculate the Euclidean distances between the input vector and the weight vectors, which connect each of the neurons on the two-dimensional topology network layer with the input neurons. The second stage is to transform the distances into a new input vector by a normalized formula for training the weights between the network topology layer and the output layer. In the Model B, the first stage is to compute the distances between the winning neuron and the other neurons on the two-dimensional network topology layer according to their network coordinates, and transform them into a new input vector by the normalized formula for training the weights between the network topology layer and the output layer. In the same way, the coordinate distances for the other neurons can be generated, and then transformed into a new input vector by the normalized formula for training the weights between the network topology layer and the output layer in the second stage. The steps of performing mathematical calculations are shown as follows:

The First Stage: Training for the weights between the input layer and the network topology layer

【Step 1】Decide the initial values of parameters for network learning, including: learning constant η, learning decreasing constant *η_Rate*, network topology size *m*, neighboring radius *R*, neighboring radius decreasing constant R_rate, and learning cycles *N*.

【Step 2】Standardize the input vectors. Calculate the transformed values of each vector scale by the following formula.

$$X_i^{'} = \frac{X_i - X_{\min}}{X_{\max} - X_{\min}}, \quad i = 1,2,...,n$$

where X_i = the *i*'th characteristic value of the input vector X

$X_{\min}$ = minimum of the *i*'th characteristic values of all the input vectors

$X_{\max}$ = maximum of the *i*'th characteristic values of all the input vectors

$n =$ quantity of characteristic values of the input vector X

【Step 3】Find out the winning neuron. Input vector *X* of the training example, calculate the net input value of each neuron on the topology, and find out the winning neuron $Y_{(j^*,k^*)}$ with the minimum net value. That is

$$net_{(j,k)} = (\sum_i (X_i^{'} - W_{i(j,k)})^2)^{\frac{1}{2}}, \quad j,k = 1,2,...,m$$

$$net_{(j^*,k^*)} = \min_{j,k} net_{(j,k)}$$

where $W_{i(j,k)}$ represents the weight of linkage between the *i*'th input unit and the network neuron locating on the *j*'th column, the *k*'th row.

【Step 4】Calculate the output vector *Y*

IF j=j, k=k* ,Then* $Y_{(j,k)}$ *=1,Otherwise* $Y_{(j,k)}$ *=0*

【Step 5】Calculate the adjusted weight ΔW for each connection weight

$$\Delta W_{i(j,k)} = +\eta * (X_i - W_{i(j,k)}) * R\_factor_{(j,k)}$$

$$i=1,2,\dots,n$$

$$j,k=1,2,\dots,m$$

In this research, the *Mexican hat function* is regarded as the *neighborhood activation function*

$$R\_factor_{(j,k)} = \exp^{\frac{-r_{(j,k)}}{R}}$$

where $r_{(j,k)}$ is the topology distance between the neuron $Y_{(j,k)}$ and the winning neuron $Y_{(j^*,k^*)}$ in the network topology layer. Its formula can be expressed as

$$r_{(j,k)} = [(j - j^*)^2 + [(k - k^*)^2 \;]^{1/2}$$

【Step 6】Update the weight matrix *W*

$$W_{i(j,k)} = W_{i(j,k)} + \Delta W_{i(j,k)}$$

【Step 7】Repeat Step 3 to Step 7 until all the training examples have been trained.

【Step 8】Reduce the learning constant η and neighboring radius *R*

$$\eta = \eta\_rate * \eta$$

$$R = R\_rate * R$$

【Step 9】Repeat Step 3 to 8 until all predetermined learning cycles are completed or come to a convergence.

The Second Stage: Training for the weights between the network topology layer and the output layer

Fix the weights between the first two layers after the training is finished in the first stage. During the training process in the second stage, the Euclidean distances between the input vector and the weight vectors (for Model A) or the topology distances between the winning neuron and all the other neurons on the network topology layer (for Model B) have to be calculated first, and data standardization is then implemented as the input values for training the weights between the second and third layer.

【Step 1】Decide the initial values of parameters for network learning.

(Step 2) Calculate the output values from the network topology layer

1. For Model A

Calculate the Euclidean distance between the input vector and the neurons on the network topology layer for the training example using the following formula

$$L_{(j,k)} = \| X(C) - W(C_{j,k}) \| = \sqrt{\sum_i [X_i(C) - W_i(C_{j,k})]^2}$$

j,k=1,2,…m

where $X(C)=$ the input vector for the training example *C*

$W(C_{j,k})=$ the weight vector connecting neuron (j,k) on the second layer with the input neurons

$X_i(C)=$ the *i*'th characteristic value of the input vector for the training example *C*

$W_i(C_{j,k})= W_{i(j,k)}$ = the weight connecting neuron (j,k) on the second layer with the *i*'th neuron on the input layer.

2. For Model B

Find out the coordinates of the winning neuron (j^*,k^*), and calculate the network topology distances from the other neurons. The formula can be expressed as

$$net_{(j^*,k^*)} = \min_{j,k} net_{(j,k)}$$

$$L_{(j,k)} = \sqrt{(j - j^*)^2 + (k - k^*)^2} \qquad j,k = 1,2,...,m$$

(Step 3) Standardize the network topology distances obtained from Step 2 using the following formulas

$$Y_{(j,k)} = 1 - \frac{L_{(j,k)} - L_{\min}}{L_{\max} - L_{\min}}, \qquad j,k = 1,2,...,m$$

$$L_{\min} = \min_{j,k}\{L_{(j,k)}\}$$

$$L_{\max} = \max_{j,k}\{L_{(j,k)}\}$$

(Step 4) Input the vector $Y_{(j,k)}$ for training the weights connecting the neurons on the second layer with the output layer. The training steps and formulas are the same as those in the first stage.

4. Evaluation of Model Efficiency

The main goal of the 3LSOM developed in this research is to implement clustering analysis through the network model using the characteristics of the collected data. To evaluate the efficiency of the proposed model, two test examples, four groups of the part images and the part/machine incidence matrix about the production processes, are adopted in the experiment.

4.1 Clustering for the Part Images

Clustering analysis involves grouping the similar parts into part families according to some characteristics of parts, like shape and size. Because of the similarity of the characteristics in the same part family, the parts of each family can be produced by those specific machine tools that have already been grouped, as well as the common molds and clamping apparatus. Therefore, manufacturing with the equivalent or similar methods is feasible, and reduces not only the preparation time of parts but also the labors, time, and expense of changing the clamping apparatus. A set of 16 part images used in the study of *Kamarthi, et al.* (1990) is employed to verify the feasibility of our developed model [8]. The training of the 3LSOM model is conducted depending on the given result, that the part images can be divided into four clusters, in order to test the efficiency of the enhanced clustering network.

Each parameter of network training in this study is set as the follows [1]:

The First Stage: the parameter settings from input layer to network topology layer

1. Input layer: 80 input neurons
2. Network topology: 10 x 10 network
3. Learning constant: 0.2
4. Learning decreasing constant: 0.95
5. Initial neighboring radius: 9
6. Neighboring radius decreasing constant: 0.95
7. Initial connection weights: Randomly assigned from –0.3 to 0.3.
8. Learning cycles: 1000

The Second Stage: the parameter settings from network topology layer to output layer

1. Input layer: 100 neurons on the network topology in the first stage as the input neurons
2. Output layer: 10 x 10 output neurons.
3. Other parameters are the same as the first stage.

To evaluate the clustering capability of clustering of the 3LSOM, a noisy signal method is adopted to obtain the test data. In this study, 5% is regarded as the first noisy signal ratio. Then, it is increased by 5% each time until 30% to get the input data for identifying the clustering capability of the proposed models and compare the results with those from three different models, FSOM, FCM and K-Means, shown in Table 1.

The evaluation of clustering capability of the 3LSOM is measured by the accuracy rate, which is the percentage of the parts that have been clustered correctly to the given result, and the formula and results are shown below:

$$\text{Accuracy rate} = \frac{\text{number of parts clustered correctly}}{\text{totoal number of parts}} \times 100\%$$

Table 1. The averages of accuracy rate of various clustering models

Degree of Noisy %	*3LSOM*	*FSOM*	*FCM*	*K-Means*
0	100	100	80.675	100
5	100	98.653	71.525	100
10	93.75	87.791	64.574	97.989
15	87.50	75.082	53.186	90.021
20	83.75	62.507	53.688	83.227
25	73.75	56.259	53.131	70.255
30	63.75	52.254	53.425	61.071

Note: The accuracy rates for FSOM (Fuzzy SOM), FCM (Fuzzy C-Means) and K-Means are adopted from Kuo et al. (2001).

Conclusions from the testing results are described as the follows:

1. The stepwise 3LSOM has the capability of clustering and mapping the same group of data points onto the same neuron on the output layer.
2. There is no significant difference between the results from the 3LSOM and FSOM in resisting noise in the input data, but the 3LSOM is easier in the judgment of group assignment for the input data than the FSOM.
3. It is necessary to give a cluster quantity in advance during clustering for both FCM and K-Means, and they are not capable of clustering automatically. In other words, the K-Means has a better ability to deal with noisy signals if the cluster quantity is already known. However, it is usually not easy to predetermine the cluster quantity in most real cases. For this reason, the proposed 3LSOM proves to be a much better clustering tool for automatic clustering automatically, and it meets practical applications.
4. For the stepwise 3LSOM, the total distance in the second stage is smaller than that in the first stage, showing that the speed of convergence in the second stage is much faster than that in the first stage.
5. The quantity of neurons on the output layer of the 3LSOM has no obvious influence on the clustering results.

4.2 The Part/Machine Incidence Matrix

In this section, the stepwise 3LSOM is employed to cluster the part/machine incidence matrix of production processes information in this section. The data are taken from Chi et al. [3]. There are 23 machines and 30 parts with multi-process-plan in the production processes. The best combination of the production paths (or flows) is adopted and is regarded as the input data for clustering analysis in this study. The final result can be shown as Table2. The parameters for the 3LSOM can be expressed as follows.

The first stage: The parameter settings from the input layer to the network topology layer

1. Input layer: 23 input neurons
2. Network topology: 10 x 10 network
3. Learning constant: 0.4
4. Learning decreasing constant: 0.95
5. Initial neighboring radius: 9
6. Neighboring radius decreasing constant: 0.95

7. Initial connection weights: Randomly assigned from –0.3 to 0.3.
8. The learning cycles: 1000

The second stage: The parameter settings from the network topology layer to the output layer

1. Input layer: 100 neurons on the network topology are regarded as the input neurons
2. Output layer: 10 output neurons for one-dimensional network topology, and 10×10 output neurons for two-dimensional network topology
3. Other parameters are set as the same as those in the first stage.

The results are depicted in Figure 2 to 5.

0	6	0	1	0	0	8	6	0	7	2
	0	1	2	3	4	5	6	7	8	9

Figure 2. The result of clustering on one-dimensional network topology for Model A

9	9		8	6						
8										
7										
6										
5										
4									5	
3										1
2										
1						1				
0										
	0	1	2	3	4	5	6	7	8	9

Figure 3. The result of clustering on two-dimensional network topology for Model A

0	2	7	0	0	0	0	6	8	0	7
	0	1	2	3	4	5	6	7	8	9

Figure 4. The result of clustering on one-dimensional network topology for Model B

9										
8				9	6		7	2		
7										
6										
5										
4										
3		4	2							
2										
1										
0										
	0	1	2	3	4	5	6	7	8	9

Figure 5. The result of clustering on two-dimensional network topology for Model B

As can be seen, the clustering results obtained by the proposed 3LSOM are the same as those from Chi et al. [3]. In addition, the results show that it can cluster the similar data into the same point or neighboring points on the output layer effectively. According to the application results and comparisons with previous researches, it is obvious that the 3LSOM has a significant capability of differentiating efficiently the difference among the input data, thus improving the usefulness and clustering accuracy of the input data has been improved.

5. Conclusion

The proposed three-layered self-organizing map neural network, 3LSOM, aims to improve the drawback of the conventional SOM in the determination of cluster number and data assignment by human judgment on the basis of the mapping result depicted on the two-dimensional network topology. To generate the inputs for training the connection weights between the network topology layer and the output layer, two data manipulation models are used and compared before being selected for the applications. To verify the feasibility and accuracy of the 3LSOM, the clustering result has been compared with those obtained using the FSOM, FCM and K-Means. The accuracy achieved by the 3LSOM is better than that by the FSOM or FCM; however, it is a little lower than that obtained by the K-Means. In addition, according to the results from the applications, this research obtains some remarkable conclusions as the follows.

1. The 3LSOM has a better capability to distinguish the differences among the input data, and increases the availability and correctness (or accuracy) effectively.
2. The image data of work parts adopted in this research are provided with obvious features for verifying the feasibility of the proposed 3LSOM. This application is quite easy to be clustered by the SOM-related models. However, the high-accuracy result may not be obtained when the input data become too complicated for analysis. Therefore, further applications should be implemented to ensure the effectiveness of the 3LSOM.
3. As seen in the results, the neighboring data points (or neurons) on the network topology layer can be clustered and mapped to the same or neighboring neurons on the output layer through the 3LSOM. It means that those data belonging to the same cluster can be located on the same location evidently without toilsome human judgment.
4. Data mining has been a popular topic in recent years. This research focuses mainly on data cluster analysis, which might be the initial part of data mining techniques for some problems. For this reason, the future studies are recommended to further analysis in order to mine out the useful rules after clustering process.
5. Other models like ART-related neural networks are recommended to replace SOM between topology layer and output layer.

Acknowledgements

The authors would like to thank the National Science Council of the Republic of China for the funding support of the related research project: NSC 92-2212-E-211-010.

References

[1] Chi, S.C., R.J. Kuo and P.W. Teng, “A Market Segmentation Approach through Fuzzy Self-Organizing Map Neural Network and Fuzzy c-Means Algorithm,” Accepted by Fuzzy Sets and Systems, 2003.

[2] Kuo, R.J., S.C. Chi, and P.W. Teng, "Generalized Part Family Formation through Fuzzy Self-organizing Feature Map Neural Network,” Computers and Industrial Engineering, Vol.4, pp.79-100, 2001.

[3] Chi, S.C., I.J. Lin and M.C. Yan, "A Granular Genetic Algorithm for Machine Cell Formation," Proceedings of the 7th International Conference on Systemics, Cybernetics and Informatics, pp.142-147, Orlando, 2003.

[4] Alahakoon, D., S.K. Halgamuge and B. Srinivasan, "Data Mining with Self Generating Neuro-Fuzzy Classifiers," IEEE International Fuzzy Systems Conference Proceedings, pp.1096-1101, August 22-25, 1999.

[5] Babu, G.P. and M.N. Murty, "Simulated Annealing for Selecting Initial Seeds in the K-Means Algorithm," Industrial Journal of Pure Applied Mathematics, Vol.25, pp.85-94, 1994.

[6] Bhuyan, J.N., V.V. Raghavan and V.K. Elayavalli, "Genetic Algorithm for Clustering with an Ordered Representation," in Proceedings of the 4th International Conference on Genetic Algorithms, San Mateo, CA: Morgan Kaufman,1991.

[7] Vesanto, J. and E. Alhoniemi, “Clustering of the Self-Organizing Map,” IEEE Transactions on Neural Network, Vol.11, No.3, May 2000.

[8] Kamarthi, S., S.T. Kumara, F.S. Yu and I. Ham, “Neural Networks and their Applications in Component Design Data Retrieval,” Journal of Intelligent Manufacturing, Vol. 1, pp.125-140, 1990.

[9] Klein, R.W. and R.C. Dubles, "Experiments in Projection and Clustering by Simulated Annealing," Pattern Recognition, Vol.22, pp.213-220, 1989.

[10] Krishnapuram, R. and J.M. Keller, "A Possibilistic Approach to Clustering," IEEE Transactions on Fuzzy Systems, Vol. 1, pp. 98-110, May 1993.

[11] Krishnapuram, R. and J.M. Keller, "The Possibilistic C-Means Algorithm: Insights and Recommendations," IEEE Transactions on Fuzzy Systems, Vol. 4, pp. 385-393, August 1996.

[12] Krishna, K. and M.M. Narasimha, "Genetic K-Means Algorithm," IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol.29, No.3, pp.433-439, June 1999.

[13] Punj, G. and D.W. Steward, "Cluster Analysis in Marketing Research: Review and Suggestions for Applications," Journal of Marketing Research, Vol.20, pp.134-148, 1983.

[14] Kuo, R.J., L.M. Ho and C.M. Hu, "Integration of Self-Organizing Feature Map and K-Means Algorithm for Market Segmentation," Computers and Operations Research, Vol.29, pp.1475-1493, 2002.

[15] Selim, S.Z. and Ismail, M.A., "K-Means-Type Algorithms: a Generalized Convergence Theorem and Characterization of Local Optimality," IEEE Transactions on Pattern Analysis and Machine Intelligent, Vol. 6, pp.81-87, 1984.

[16] Sharma, S., Applied Multivariate Techniques, John Wiley, pp.212-216, 1996.

[17] Zamir, O., O. Etzion, O. Madani and R.M. Karp, “Fast and Intuitive Clustering of Web Documents,” Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, pp. 287-290, 1997.

Table 2. The final clustering result for the part/machine incidence matrix with 30 parts and 23 machines

N \ M	14	12	3	22	17	8	11	6	4	18	23	16	15	10	1	9	20	21	5	13	2	19	7
5a	1	1			1	1																	
6a	1		1	1																			
9a	1		1	1	1	1																	
13a		1	1	1	1	1																	
19b	1	1	1		1																		
22a	1	1	1	1	1	1																	
29b		1	1		1	1			1														
1b							1	1	1		1	1											
4b									1	1	1	1											
10a							1	1	1	1						1	1						
15b							1	1	1	1													
18a							1		1	1	1	1											
20a							1		1	1	1	1											
21a									1	1	1	1											
23c						1	1	1		1		1											
26a							1	1		1	1												
2a													1		1		1	1			1		
7b													1	1			1	1					
11b													1	1	1		1						
16c													1		1	1							
17a													1	1	1	1	1	1					
25a													1	1		1	1						
28c															1	1	1	1					
30c													1			1	1	1					
3b																			1	1	1	1	1
8b																			1		1	1	1
12b																			1	1	1		1
14a													1						1	1	1	1	
24b																			1	1		1	1
27c																				1	1	1	1

10