A Clustering Algorithm Using the Ordered Weight Sum of Self-Organizing Feature Maps

Jong-Sub Lee[1] and Maing-Kyu Kang[2]

[1] Department of Technical Management Information Systems, University of Woosong, Daejeon, South Korea
ljs@wsu.ac.kr
[2] Department of Information & Industrial Engineering, University of Hanyang, Ansan, South Korea
dockang@hanyang.ac.kr

Abstract. Clustering is to group similar objects into clusters. Until now there are a lot of approaches using Self-Organizing Feature Maps(SOFMs). But they have problems with a small output-layer nodes and initial weight. This paper suggests one-dimensional output-layer nodes in SOFMs. The number of output-layer nodes is more than those of clusters intended to find and the order of output-layer nodes is ascending in the sum of the output-layer node's weight. We can find input data in SOFMs output node and classify input data in output nodes using the Euclidean Distance. The suggested algorithm was tested on well-known IRIS data and machine-part incidence matrix. The results of this computational study demonstrate the superiority of the suggested algorithm.

1 Introduction

Clustering deals with data corresponding to processes of clusters. The predicament of clustering is the input of *n* data of multi dimension and dividing it into *k* cluster having similar features. When compared with common data, data in a cluster has greater similarities rather than the differences. The measurement of the similarity is calculated by the Euclidean Distance Method based on the Attribute Value of Data. The shorter the size of Euclidean distance, the higher the resemblance.

Clustering Algorithms can be divided into two categories; hence Hierarchical Algorithms and Partition Algorithms. Recently, there has been a dynamic study of clustering algorithms utilizing neural network and fuzzy-neural networks The hierarchical algorithm, which is postulated by Mangiameli *et al*[10], depicts that there is single linkage clustering which measures the similarity by the minimum distance between two clusters, complete linkage clustering which measures the similarity by the maximum distance between two clusters, group-average clustering which measures the similarity by the average distance between the two clusters, and finally consistent with Ward's Hierarchical Clustering measure the similarity by density between two clusters. Hierarchical Algorithms has a defect in that it cannot complete the problems caused by an inappropriate merge.

In a broader way, while we consider about the Partition Algorithms, it can be noted that such algorithms formulate partition of the data and form clusters so that data in a

M. Gavrilova et al. (Eds.): ICCSA 2006, LNCS 3982, pp. 898 – 907, 2006.

cluster is more similar than other clusters. Partition Algorithms can be characteristically classified into *k*-Means Algorithm and ISODATA Algorithm. To deal with *k*-Means Algorithm, this will incorporate the data of each cluster which is able to rearrange as the algorithms is repeated into the direction minimizing the distance difference between each data and central value of each cluster. This conquered the disadvantages of hierarchical algorithms, which was unable to overcome the inappropriate merge occurring in the early stage. ISODATA Algorithm start with *k* centrums but the number of clusters is not necessarily *k*. Despite this aspect, the number of clusters can be flexibly changed during algorithm performing. ISODATA algorithm complete *k*-Means algorithm' defect of having fixed numbers of clusters[5].

In the year of 1989, two professionals called Huntsberger and Ajjimarangsee[6] postulated a clustering algorithm modified in parameters such as learning rate and vicinity rate and slightly modified Kohonen[8]'s learning method. In 1993 another personality called Pal *et al*[11] illustrated a lose function method which provided the connecting weight of the distance between input data and output node by early connection strength, its also designated the Competitive learning neural network Algorithms that minimize the Lose function.

The fuzzy concept of neural network was divulged by two individuals named Tasao *et al*[13] and Karayiannis[7]. Particularly with Karayiannis[7] he formulated an algorithm that minimizes the connecting weight sum of square Euclidean distance between FALVQ(Fuzzy Algorithm for Learning Vector Quantization) inputting data and connecting weight of LVQ(Learning Vector Quantization).

In 2000 another person called Kusiak[9] defined the clustering problem as an NP-Complete problem. This illustrates where the number of machines in a cluster is higher, the computational complexity is exponentially increased and time consuming. To overcome such tribulations, we prefer Heuristic algorithms to Optimization algorithms.

In this erudition, the attribute of the connecting weight modifying the form into a similar one to the inputting data is consumed, when the Self-Organization neural network is in a progress of unsupervised learning with input data of Anderson's IRIS data and machine-part matrix data. The attributes of this cram depict the numbers of one dimension output nodes, learning rate and the boundary of adjacency, etc. The suggested algorithm creates a process of learning using these parameters, and groups depending on the dimensions of Connecting Weight Distance between *i* and *j* output nodes. As the result this algorithm elaborates how to reduce the numbers of misclassification.

2 SOFMs Neural Network

SOFM is a Competitive Learning Neural Network model which is explained by Kohonen[8]. Fig. 1 illustrates that SOFM includes two layers, hence an input layer formed with *m* input nodes and an output layer formed with *n* output nodes.

As the Input layer receives input data, mapping is performed at the output layer. The output layer uses either a one-dimensional structure or a two-dimensional structure. We can set the output node to either a bigger number than that of input data or a random number *k* chosen by users so that the input data can be spread on other output nodes. At each node of output, mapping is performed with the input data.

Every node of input layer and output are connected and there will be a connecting line between output node *i* ($1 \le i \le n$) and input node *j* ($1 \le j \le m$), having a Connecting

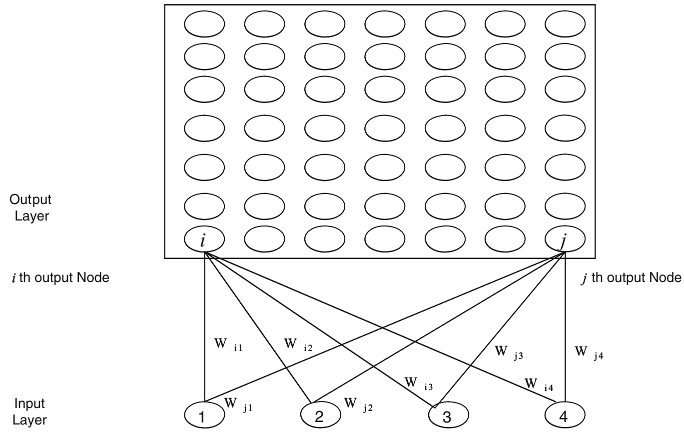

Fig. 1. General structure of SOM

Weight W_{ij}. Connecting weight is a real number between 0 and 1, which is randomly presented at the initial stage but can be changed by the input data. Corresponding with each input data, winner node i^* is selected, which is the most similar output node. As formula (1), among the distance D_i calculated between input data X and connecting weight W_i, the lowest output node i^* is selected

$$D_i = \sqrt{(x_1 - w_{i1})^2 + (x_2 - w_{i2})^2 + \cdots + (x_m - w_{im})^2} \quad i = 1, \cdots, n \tag{1}$$

Connecting weight W_{i*} of output node i^* which is given in as formula (2), will establish the nearest connecting weight to input data X among n output node. The node located on the front and the back of winner node i^* is called the Neighbor Node and Neighborhood $N_{i*}(\delta)$ is a congress of neighbor node separated as δ from winner node i^*. Winner node is specified as "#" in Fig. 2. While we refer to the other output node as "*", the neighbor node is referred with the case of which the Radius is δ=0, 1, 2 by using the Rectangular Grid.

$$| X - W_{i*} | = \min | X - W_{i*} | \quad i = 1, \cdots, n \tag{2}$$

Coherence with neighborhood as $N_{i*}(\delta)$, the connecting weight of self-organization neural network decreases the neighbor range and the learning ratio with regulating the connecting weight until the neighbor range becomes the same as the winner node itself as formula (3). The learning ratio α(t) is a ratio which is used to control the difference between input data and existing connecting weight in accordance with flow of time t. This is a real number between 0 and 1 and progressively decreases as the learning proceeds. Generally, the learning cannot be achieved accurately while the learning ratio is too high, and it takes too long when its too low[9].

Here $w(old)_{ij}$ represents the link-weight before adjustment and $w(new)_{ij}$ represents the link weight after adjustment.

$$w(new)_{ij} = w(old)_{ij} + \alpha(x_i - w(old)_{ij}) \quad i \in N_i*(\delta) \quad j = 1, \cdots, m \tag{3}$$

$$N_i*(\delta) \text{ is neighborset with } \delta \text{ far from } i*$$

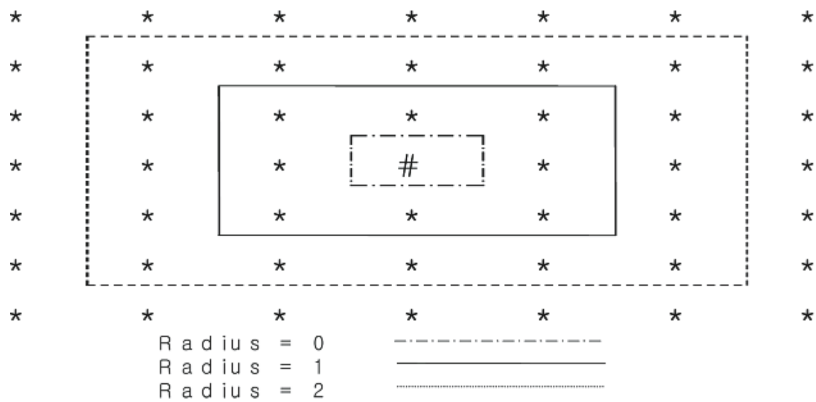

Fig. 2. The neighborhood using rectangular grid in two dimensions

Learning algorithm is summarized as follows.

1. Initialize link weight with random number between 0 and 1. Decide the range of neighbor and the learning rate.
2. Input one input vector in input layer.
3. Calculate the distance D_i between the *i*th input vector and link weight vector of formula (1).
4. Choose one winner node.
5. Adjust link weight according to learning rule of formula (3).
6. Downsize the range of neighbor and learning rate. Repeat the process from step 2 to 5 until the range of neighbor becomes winner node itself.

3 Suggestion

Taking into account at first level, the SOFM given by Kohonen has a few tribulations which are caused by the change of solution and an extensive amount of learning time. The suggested structure of SOFM formed with a one-dimensional linear output layer gets a connecting weight sum at each output node based on a randomly given connecting weight at first level. Anchored in this, it lines up in an ascending order. Matching input data to the consisted output node and transforming output node to a similar form to input node through learning, it forms a group. Exceptionally, when the number of output node is set higher than the number of input data, it enables the distribution of the output node to be spread in similar form as the distribution of input data. During the learning process, at the point of neighbor range's being half at the first level, the connecting weight of output node that had not been winner node is not regulated as formula (3). In accordance with the same situation, when learning processed input data is lined up on the output node which has a similar order, the required group can be formed by linear separating the highest point of the connecting weight among output nodes. This disentangles the defect which can occur in learning process, and apparently acts upon the process of forming a group after learning.

Suggested clustering algorithm settles on the parameter such as a number of output node, learning ratio, and neighbor range. Smaller the number of output node, shorter the learning time. Least number of output node enables the input data to adequately spread on the output node. According to practical method, the best solution is shown by how to trounce the drawbacks which are possibly caused from the learning process; it would be gained if the number of output node is set to more than twice of the number of input data. As the learning ratio gets higher, the learning time decreases. At first level, set every output mode as neighbor, diminish the range as time passes, and stop the algorithm when the neighbor range is itself. Set the initial learning ratio, $\alpha(0)$, to 0.4 which practically offers good results, and set the initial neighbor range as the same number as that of output node.

With the assistance of Euclidean distance, separate the output node i, on which each input data is mapped from the distance between the connecting weight i+1. Where i and i+1 represent the number of neighboring output node. That is why the connecting weight consists of a similar form of probability distribution function; the mapped neighboring output node's connecting weight is used as Euclidean distance. So the neighboring output node's connecting weight simply needs to be considered.

Suggested Clustering algorithm process is as follows,

1. Initialize the structure of SOFM (output node type, number of input or output node)
2. Initialize the weight on each connecting line, and set an initial learning ratio $\alpha(0)$ and learning ratio function $\alpha(t)$, and an initial neighboring range.
3. Gain the sum of connecting ratio at each output node. Depends on the volume of connecting weight sum, arrange the output node with an ascending order.
4. As formula (1), refers to each input data, calculate the distance between the connecting weight of output node and input data and decide the winner node which is the shortest node.
5. Regulate the connecting ratio of neighbor node located at a regular range from the winner node.
6. Decrease the neighbor range as 1 and learning ratio as $\alpha(t)=(1\text{-}t/4950)\times\alpha(t\text{-}1)$. Finally, repeat Step 4 to 5 until the neighbor range becomes winner node itself or learning ratio to be 0. Exclude the tip that the neighbor range becomes half of the initial neighbor range, hence any disastrous output node becomes a winner node and thereafter learning is not required.
7. Map each input data on the nearest output node.
8. Calculate the connecting weight distance WD(i, j) between the neighboring output node i and j of which the input data is matched.
9. At the point of an output node with linear structure, selecting k-1 (output node range) which has the greatest difference of connecting weight distance WD(i, j) is able to form k group.

4 Numerical Example

In this premise, we recycled the postulations presented by Anderson's IRIS data: [1] and machine-part matrix: [2],[3],[4],[9],[12]. Anderson's IRIS data, especially, consists of 150 samples of data which have parameters of 4 dimensions (Petal Width, Petal Length, Sepal Width, and Sepal Length). The three clusters (Iris Setosa, Iris Verisiclolr, Iris Verginica) consist of 50 pieces of data each. It is generally noted that IRIS data, as a clustering algorithm which devours unsupervised learning, is known to fabricate about 15 to 17 misclassifications[11].

Following are some distinctive applications of suggested clustering algorithms to IRIS date used for this study.

1. The structure of SOFM used for this example is depicted as is in the subsequent case. The structure of the output node is linear and the number of the input nodes and the output nodes are 4, 300 each. In this situation, the structure of the output nodes is also linear, and the number of the output doubles the input data. Hence, the number of input nodes is 4.

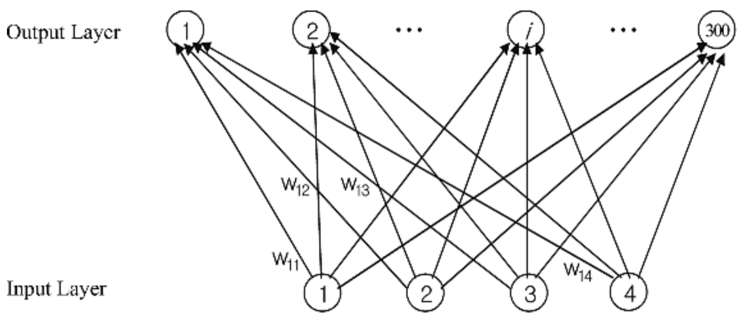

Fig. 3. Suggesting structure of SOFMs

2. The weight of the first output node is W_1={0.5882, 0.2500, 0.2200, 0.1872}, the weight of the second output node is W_2={0.9232, 0.4950, 0.8613, 0.7534}, the weight of nineteenth output node is W_{19}={0.0145, 0.2887, 0.0584, 0.0835} and the weight of 300th output node is W_{300}={0.7083, 0.8935, 0.8302, 0.0759}. Set the initial learning ratio as 0.4, and the learning function is $\alpha(t)=(1-t/4950)\times\alpha(t-1)$.
3. The sum of the weight of the output nodes: the first: 1.245, the second: 2.5468, the 19th :0.4453, the 300th: 2.5082. The order by size is this: 19, 48,…, 44, 89.
4. When calculating the distance W_{19}, the weight of the first output to the first input data as formula (1), it produces 0.6902, 1.3757, 1.6861, … . And the shortest output node, W_{19}, is called the winner node.
5. Adjust the weight of all output nodes within 300 radiuses as formula (3).
6. Reduce the neighbor rate one by 1, and in the case when *t* is zero, reduce the initial learning ratio $\alpha(0)$ to 0.4, and in the case when *t* is 1, reduce it as $0.4\times(1-(1/4950)$ and repeat the Step 4 to 5 until the neighbor rate becomes the winner node itself.
7. When mapping each data to the nearest output node, it is as Table 1.
8. When calculating the distance of each output node and the weight distance, the results are as follow: WD(1, 5)=0.011, WD(5,8)=0.0074, WD(8,10)=0.1738, WD(10,12)=0.0286, ... ,WD(56, 75)=2.8132, … , WD(182, 191)=0.4176, … , WD(299, 300)=0.0139.

Table 1. The Data in the output nodes

No[1]	Data[2]	No	Data	No	Data	No	Data	No	Data
1	6,11	41	14,43	100	90	169	59	241	133
	15,16	42	39	102	63	170	55	245	116,149
	17,19	44	4,9,	105	83,93	175	51,77,87	250	142,146
	33,34		13	111	68	176	53,57	253	137
5	37,49	47	2,10	114	107	182	78	260	105
8	20,45		46	115	91	191	73,134	261	101
	47	48	42	116	95	196	124,127	262	141,145
10	21	49	35	119	100	202	128,139	263	113
12	22,32	50	30	121	89	204	71	264	140
15	28	52	31	125	97	205	120	266	125
18	18	54	26	126	85,96	206	84	267	109
19	1,5,	56	25	128	56,67	211	150	271	121
	29,44	75	99	134	62	213	122	272	144
21	24,40	77	58	136	72	214	102,114	280	103
22	27,41	78	94	144	79,86		143,147	283	130
23	8,38	80	61	147	98	218	115	284	126
27	50	83	80	150	69,88	219	135	286	110
29	12	85	82	151	74	227	104	293	131
31	36	89	81	153	92	232	138,148	294	108
32	23	93	65	154	64	236	112	299	136
34	7	96	70	157	75	237	117	300	106,118
39	3	97	54	166	52,76	238	111		119,123
40	48	98	60	167	66	240	129		132

Table 2. The Data in 3 Groups

No of Group	No of Input Data
1	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, 44, 45, 46, 47, 48, 49, 50
2	51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 72, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 107
3	71, 73, 84, 101, 102, 103, 104, 105, 106, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150

9. The output node and the section of output node that have the longest weight distance are 56 and 75 and the next are 182 and 191. Each value is WD(56, 75)=2.8132, WD(182, 191)=0.4176. Therefore, the 3 groups are same as table 2.

---

[1] No of Output Node.
[2] No of Input Data.

5 Result

The suggested revision is an assessment on IRIS data[1], referred by existing Studies and machine-part matrix data. As Table 2 shows, the misclassified data appear one in group 2, three in group 3 and the total is four. This analysis created a better consequence than the existing algorithms that appears in Table 3.

The recommended structure of SOFM of the output node is one-dimensional linear, nevertheless as the initial arbitrarily set weight; the sum of weight of the output node was not able to line up from left to right by its magnitude. To make it acceptable, we lined them up setting up the sum of weight as a standard from left to right in ascending order. In addition to this, for fixing on the number of output nodes which is probable to get the nearest optimizing quotient of IRIS data and machine-part matrix, we have carried out the experiment by putting the number of output nodes from 3 as a start, and increased it to quadruple input data. As a result, we found that the results under the situation of 3 output and less than the double of input data were insignificantly different from time to time; however, the result under the situation of more than two times of input data were the same.

The initial neighbor range is set as 300 in the radius, so as to enable amending the weight of all output nodes. Once the learning process passed over half of the initial neighbor range, the output nodes failed to become winner nodes, hence cease adjusting the weight.

Table 3 shows the result of measure up to the quotients of the suggested clustering algorithms to Anderson's IRIS data and of existing algorithms.

It can be widely known that the suggested clustering algorithms accomplished the quotient with 4 misclassifications, which is a enhanced result from Pal *et al*[11]'s 17 misclassifications and Karayiannis[7]'s 15 misclassifications.

According to Pal *et al*[11], the existing clustering algorithms that use unsupervised learning produce at least 15 to 17 misclassifications. The suggested clustering algorithms produce only 4 misclassifications which is lesser than presently existing algorithms.

The suggested clustering algorithms used the same parameter to solve the machine-part grouping problem which is well known in manufacturing field. When applied to IRIS data that has a value of 4 dimensional real number, setting the initial learning ratio as 0.4, and the learning function (1-t×(1/4950). The machine-part grouping problem consists of the machine-part matrix, which has no exceptional elements.

Table 3. The number of error Comparison for Anderson's IRIS Data

Source of Problem	Source of Algorithms	No of Error
Anderson's IRIS Data Set	Suggested algorithm	4
	Karayiannis[7]	15
	Pal *et al*[11]	17

In the initial phase of machine-part grouping, Table 4 convinced the optimizing number of group and misclassifications that may occur during the grouping. The suggested clustering algorithm forms the machine cells, utilizing an independent machine-part matrix, and indicates the number of machine cells and misclassifications that will occur during the process, as Table 4. The suggested clustering algorithms indicate the optimizing number of machine cells and the minimum number of classifications, 0.

Table 4. The number of error Comparison for Machine-Part Incidence Matrix

Size[3]	Source of Problems	No of Group		No of Error	
		Optimal	Suggested Algorithm	Optimal	Suggested Algorithm
4×5	Kusiak[9]	2	2	0	0
10×15	Chan *et al*[2]	3	3	0	0
10×20	Srinivasan *et al*[12]	4	4	0	0
24×40	Chandraseharan *et al*[4]	7	7	0	0
40×100	Chandraseharan *et al*[3]	10	10	0	0

6 Conclusion

This revise contributes an effective clustering algorithms classifying the IRIS data, and forming the machine-part groups. The features portrayed here are the structure of SOFM studying parameter. The structure of SOFM suggested in a one-dimension is linear. The number of output nodes is set as twice as the number of input node.

We endowed the weight to the output node voluntarily, and arranged them as the size of the sum of weight in ascending order. Once the learning is in process, and the neighbor range arrives at the point of half of the initial neighbor node, the weight of output node that has failed to be the winner node stops adjusting the weight. Analogous with output node the input data learning is completed and takes its turns, it is possible to form a group by linear dividing the point that has the largest difference of weight between output nodes.

According to the experienced method, to set the number of output node to more than twice the number of input data is the best way to achieve the best quotient that is able to overcome the defects that might occur during the learning process. In the neighbor arrangement, it set all output nodes as its neighbor, and as time goes by, it reduces the range, and finally when it comes to be its own neighbor, the algorithms stops.

A well recognized method called IRIS data and machine-part matrix is used in this premise. As the result of research on IRIS data, we achieved better quotient with only 4 misclassifications. But the normal way of the existing clustering algorithms utilizing unsupervised learning produces 15 to 17 misclassifications. So, broadly saying, the suggested algorithms do not use a complex operation, hence the suggested algorithms perform more flexibly and feasibly in real time applications.

[3] No of Machine× No of Part.

References

1. Anderson, E.: The IRIS's of the Gaspe Peninsula, Bull. Amer. IRIS Soc., 59 (1939) 2-5
2. Chan, H. M., Milner, D. A.: Direct clustering algorithm for group formation in cellular manufacturing, Journal of Manufacturing Systems, 1(1) (1982) 65-75
3. Chandrasekharan, M. P., Rajagopolan, R.: ZODIAC: An algorithm for concurrent format of part-families and machine-cells, International Journal of Production Research, 25(6) (1987) 835-850
4. Chandrasekharan, M. P., Rajagopolan, R.: Groupability: An analysis of the properties of binary data matrices for group technology, International Journal of Production Research, 27(6) (1989) 1035-1052
5. Everitt, B. S., Laudau, S., Leese, M.: Cluster Analysis, Edward Arnold, London (2001)
6. Huntsberger, T. L., Ajjimarangsee, P.: Parallel Self-Organizing Feature Maps for Unsupervised Pattern Recognition, International Journal of General Systems, 16(4) (1990) 357-372
7. Karayiannis, N. B.: A Methodology for Constructing Fuzzy Algorithms for Learning Vector Quantization, IEEE Trans. Neural Networks, 8(3) (1997) 505-518
8. Kohonen, T.: Self-Organizing Maps, Springer, Berlin (1997)
9. Kusiak, A.: Computational Intelligence in Design and Manufacturing, John Wiley & Sons, New York (2000)
10. Mangiameli, P., Chen, S. K., West, D.: A Comparison of SOM Neural Network and Hierarchical Clustering Methods, European Journal of Operational Research, 93(2) (1996) 402-407
11. Pal, N. N., Bezdek, J. C., Tasao, E. C. -K.: Generalized Clustering Networks and Kohonen's Self-Organizing Scheme, IEEE Trans. Neural Networks, 4(4) (1993) 549-551
12. Sirnivasan, G., Narendran, T.T., Mahadevan, B.: An assignment model for the part families problem in group technology, International Journal of Production Research, 28(1) (1990) 145-152
13. Tasao, E. C. -K., Bezdek, J. C., Pal, N. N.: Fuzzy Kohonen Clustering Networks, Pattern Recognition, 27(5) (1994) 754-757