

3M Algorithm: Finding an Optimal Fuzzy Cluster Scheme for Proximity Data

Ying Xie¹, Vijay. V. Raghavan¹, Xiaoquan Zhao²

¹The Center for Advanced Computer Studies
University of Louisiana at Lafayette
Lafayette, LA 70504-4330
{yingxie, raghavan}@cacs.louisiana.edu
²GE Medical Systems
Xiaoquan.Zhao@med.ge.com

Abstract - In order to find an optimal fuzzy cluster scheme for proximity data, where just pairwise distances among objects are given, two conditions are necessary: A good cluster validity function, which can be applied to proximity data for evaluation of the goodness of cluster schemes for varying number of clusters; a good cluster algorithm that can deal with proximity data and produce an optimal solution for a fixed number of clusters. To satisfy the first condition, a new validity function is proposed, which works well even when the number of clusters is very large. For the second condition, we give a new algorithm called multi-step maxmin and merging algorithm (3M algorithm). Experiments show that, when used in conjunction with the new cluster validity function, the 3M algorithm produces satisfactory results.

I. INTRODUCTION

Proximity data, where just pairwise distances among objects are given, are often encountered in fields such as image recognition, document retrieval and spatial data analysis. Finding an optimal fuzzy cluster scheme for proximity data is an important task in those fields and this problem is not yet well solved. Although a few clustering algorithms such as k-medoids [2], subtractive clustering [7] and maxmin method [13] can be applied to proximity data, as is the case with other clustering approaches used when feature vectors associated with various objects are known [1,6,8,9,19], they suffer from the following difficulties:

1) The number of clusters, which is one of the most important factors that determine clustering quality, depends on user's input parameter, such as the expected number of clusters, the minimal density value or the threshold coefficient of the distance between cluster centers. Thus, it is difficult to guarantee that the clustering result can reflect the natural cluster structure of the data sets. Generally there are two mechanisms to solve this problem. One mechanism consists of a merge approach, such as that used by compatible clustering merging algorithm [10] and extended fuzzy-c means [5]. They begin the clustering process with a large number of clusters, and gradually reduce the number by merging the most compatible or similar pairs of clusters until a specified merging criterion is no longer satisfied. For this kind of approach, the final number of clusters is always sensitive to one or two user-selected parameters that define the threshold criterion for merging. Though they apply some

compatibility or similarity measure to choose the clusters to be merged, no validity measure is used to guarantee the clustering results after a merge are better than the one before the merge. The other mechanism applies a validity measure to obtain a suitable number of clusters. For this purpose, several cluster validity measures have been proposed. In [11], X. L. Xie and G. Beni report that fuzzy clustering validity function proposed in [12] has monotonic decreasing tendency with the number of clusters; the hard partition validity criterion proposed in [11] always decreases as the number of clusters is increased. Through experiments, we find that both Partition Entropy and Average Partition Separability [12] tend to go up as the number of clusters increases. Thus, those measures are not suitable to be used as the goodness index to determine the number of clusters. X. L. Xie and G. Beni also proposed a compactness-separation validity index S [11], which is considered to be independent of the number of clusters [4]. This index is given by

$$S = \frac{\sum_i^c \sum_j^N \mu_{ij}^2 \|r_i - x_j\|^2}{N \times \min_{i,j} \{\|r_i - r_j\|^2\}}$$

where c is the number of clusters; N is the number of objects; r_i is the center of i th cluster; x_j is the j th object; and μ_{ij} is the membership value of the j th object to the i th cluster.

As we can see, this validity index uses the minimum distance between the cluster centers as the measure to evaluate the separation of the clusters, which does not seem reasonable. Two clustering schemes for a certain data set, even though having the same minimum distance between the centers of clusters, may have very different degrees of separation. Another problem about this validity function is that S tends to monotonically decrease when c is very large [11].

2) All these methods are sensitive to some initial parameters. For example, k-means, fuzzy c-means and k-medoids may give very different clustering results with different initial partition. Some density parameter and noise threshold may significantly influence the clustering quality of subtractive clustering as well as other density-based approaches [15]. Maxmin, a very simple method that always tries to make the clusters as separate from each other as possible, also may produce different clustering schemes with

different start points. Thus, even though the number of clusters is given, these methods cannot guarantee that an optimal solution will be obtained.

Therefore in order to obtain optimal fuzzy cluster scheme for proximity data, two conditions are necessary:

- (a) A good validity function, which can be applied to proximity data for the evaluation of the goodness of cluster schemes for varying number of clusters.
- (b) A good cluster algorithm that can deal with proximity data and produce an optimal solution for a fixed number of clusters.

Once these two requirements are met, the strategy of getting the optimal fuzzy cluster scheme is easy: produce the optimal solution for each potential number of clusters, and use the validity function to choose the best one, thus automatically deciding on the number of clusters.

To satisfy the first condition, we propose a new validity function to evaluate the compactness and separation of different clustering results for a given proximity data set. Experiments show that, unlike the situation when index S is used, even when the number of clusters is very large, our validity function still works well. For the second condition, we propose a new fuzzy clustering algorithm---*Multi-step Maxmin and Merging algorithm (3M algorithm)* to deal with proximity data. The 3M algorithm consists of two components. The first one, which is called *multi-step maxmin*, extends the basic maxmin method with some optimization steps. Maxmin method, which tries to make the cluster as separate as possible, provides the initial partition. Then, the optimization process is performed in order to get the local optimum. Each center of the clusters is then used as start point, and the maxmin and optimization process are performed again. This process is repeated until the algorithm converges or a maximum number of steps is reached. Through experiments, we find that multi-step maxmin is not sensitive to the start point and can always generate very good partitions when the number of clusters is large enough. Thus, 3M algorithm always begins with a large value of number c . Assume that we have already obtained an optimal solution for a chosen number of clusters c , the purpose of the second component, the *merging algorithm* is to obtain the candidate partitions for $c-1$. Based on the candidate partitions, multi-step maxmin runs again to get the optimal solution for $c-1$. Then the newly proposed validity function is used to evaluate the goodness of solutions for each c . The cluster scheme with the largest value of validity function is reported as the final optimal solution.

The rest of this paper will be organized as follow: in section 2, a new cluster validity function is defined; we give a detailed description of 3M algorithm in section 3; some experimental results are provided in section 4; finally, in section 5, we will provide a brief conclusion.

II. A NEW VALIDITY FUNCTION

The purpose of cluster analysis is to group objects into clusters such that the association (or similarity) of objects within the same cluster is high, and it is low for the objects in different clusters. Therefore, the compactness and separation are two reasonable measures to evaluate the goodness of cluster results. Our new cluster validity function is also based on compactness and separation of clusters.

Definition 2.1 Given a cluster scheme $C = \{C_1, C_2, \dots, C_c\}$ for a proximity data set $X = \{x_1, x_2, \dots, x_N\}$, let $C' = \{C_{pi} \mid C_{pi} \in C \text{ and } C_{pi} \text{ is not singleton, } i=1, 2, \dots, k \text{ where } k=|C'|\}$, the compactness, CP, of the cluster scheme C is given by

$$CP = \frac{k}{\sum_{i=1}^k \left(\frac{\sum_{x_j \in C_{pi}, x_j \neq r_i} \mu_i(x_j)^2 d(x_j, r_i)^2}{\sum_{x_j \in C_{pi}, x_j \neq r_i} \mu_i(x_j)^2} \right)}, \quad (1)$$

where $\mu_i(x_j)$ is the membership value of x_j belonging to C_{pi} , r_i is the center of C_{pi} , c is the number of clusters and $2 \leq c < N$, $d(x_j, r_i)$ is the distance between x_j and r_i .

Definition 2.2 The separation, SP, of a cluster scheme $C = \{C_1, C_2, \dots, C_c\}$ for a data set $X = \{x_1, x_2, \dots, x_N\}$ is given by

$$SP = \left(\frac{\sum_{i=1}^c \min_{1 \leq j \leq c, i \neq j} \{d(r_i, r_j)\}}{c} \right)^2, \quad (2)$$

where c is the number of clusters, r_i is the center of i th cluster, $d(r_i, r_j)$ is the distance between r_i and r_j .

Definition 2.3 Given a cluster scheme $C = \{C_1, C_2, \dots, C_c\}$ for a proximity data set $X = \{x_1, x_2, \dots, x_N\}$, let $C' = \{C_{pi} \mid C_{pi} \in C \text{ and } C_{pi} \text{ is not singleton, } i=1, 2, \dots, k \text{ where } k=|C'|\}$, The separation-compactness, SC, of the cluster scheme C is given by

$$SC = \frac{k}{c} \times SP \times CP. \quad (3)$$

Then the objective of 3M algorithm is to find the cluster scheme which solves:

$$\max_{2 \leq c < N} \left\{ \max_{\Omega_c} \{SC\} \right\}, \quad (4)$$

where Ω_c denotes all of the candidate cluster schemes for a certain number of clusters c .

III. MULTI-STEP MAXMIN AND MERGING ALGORITHM (3M ALGORITHM)

Recall that 3M algorithm consists of two components. One is the merging algorithm, whose task is to find the candidate partitions for number $c-1$, based on the optimal solution for number c . The other is Multi-step Maxmin algorithm, which is used to find the optimal partitions at the initial stage with a large c value and after each merging process until $c \leq 2$. We will describe the merging algorithm in section 3.2, and multi-step maxmin algorithm in section 3.3. In section 3.4, the whole picture of 3M algorithm will be provided. First of all,

in section 3.1, we will briefly talk about object median and give the definition of membership function that is suitable for proximity data.

A. Object Median and Fuzzy Membership Function

For proximity data set, there is no mean value for a subset, so we use an object median [14] instead of the mean. Let C_i be a subset of a data set with distance function d , a point x_o in C_i is called a object median of C_i if

$$\sum_{y \in C_i} d(x_o, y) = \min_{x \in C_i} \left\{ \sum_{y \in C_i} d(x, y) \right\}.$$

Let $X = \{x_1, x_2, \dots, x_N\}$ be a data set, i.e., let r_j be the center of the j th cluster, $j=1, 2, \dots, c$. We define membership functions μ_{C_j} , $j=1, 2, \dots, c$, for any $x \in X$, as

$$\mu_{C_j}(x) = \begin{cases} 1 & \text{if } d(x, r_j) = 0, \\ 0 & \text{if } d(x, r_k) = 0, k \neq j, \\ \left(\sum_{i=1}^c \frac{d(x, r_i)}{d(x, r_j)} \right)^{-1} & \text{Otherwise.} \end{cases}$$

It is easy to see that

$$\sum_{j=1}^c \mu_{C_j}(x) = 1, \forall x \in X \text{ and } \sum_{k=1}^N \mu_{C_j}(x_k) \leq N, j=1, 2, \dots, c.$$

The fuzzy partition can be converted to a hard partition by choosing

$$\mu_{C_j}(x_k)_{hard} = \begin{cases} 1 & \text{if } \mu_{C_j}(x_k) = \max_{1 \leq v \leq c} \{ \mu_{C_v}(x_k) \}, \\ 0 & \text{else.} \end{cases}$$

It is easy to see that $\mu_{C_j}(x_k) = 1$ iff r_j is the nearest center of point x_k .

B. Merging Algorithm

The merge process generally used by earlier studies involves some similarity or compatibility measure to choose the most similar or compatible pair of clusters to merge into one. In our merge process, we choose the “worst” cluster and delete it. Each element included in this cluster will then be placed into its own nearest cluster. Then, centers of all clusters will be adjusted. That means, our merge process may affect multiple clusters, which we consider to be more practical. How to choose the “worst” cluster? We still use the measures of separation and compactness to evaluate individual clusters (except singleton).

Definition 3.1 Given a cluster scheme $\mathbf{C} = \{C_1, C_2, \dots, C_c\}$ for a data set $X = \{x_1, x_2, \dots, x_N\}$, for each $C_i \in \mathbf{C}$, if C_i is not a singleton, the compactness of C_i , denoted as cp_i , is given by

$$cp_i = \frac{1}{\sum_{x \in C_i, x_j \neq r_i} \mu_i(x_j)^2 d(x_j, r_i)^2} \bigg/ \sum_{x \in C_i, x_j \neq r_i} \mu_i(x_j)^2, \quad (5)$$

where $\mu_i(x_j)$ is the membership value of x_j belonging to i th cluster C_i , r_i is the center of i th cluster C_i , c is the number of clusters and $2 \leq c < N$.

Definition 3.2 Given a cluster scheme $\mathbf{C} = \{C_1, C_2, \dots, C_c\}$ for a data set $X = \{x_1, x_2, \dots, x_N\}$, for each $C_i \in \mathbf{C}$, if C_i is not a singleton, the separation of C_i , denoted as sp_i , is given by

$$sp_i = (\min_{1 \leq j \leq c, i \neq j} \{ d(r_i, r_j) \})^2, \quad (6)$$

where r_i is the center of i th cluster C_i ; r_j is the center of j th cluster C_j ; c is the number of clusters and $2 \leq c < N$.

Definition 3.3 Given a cluster scheme $\mathbf{C} = \{C_1, C_2, \dots, C_c\}$ for a data set $X = \{x_1, x_2, \dots, x_N\}$, for each $C_i \in \mathbf{C}$, if C_i is not a singleton, the separation-compactness of C_i , denoted as sc_i , is given by

$$sc_i = sp_i \times cp_i. \quad (7)$$

Thus, the “worst” cluster is the one with the least sc_i value.

Algorithm 3.1 Merging Algorithm

Input: Optimal cluster scheme $\mathbf{C}^* = \{C_1^*, C_2^*, \dots, C_{c+1}^*\}$ for a proximity data set $X = \{x_1, x_2, \dots, x_N\}$ where $c \geq 2$.

Output: Candidate cluster scheme $\mathbf{C} = \{C_1, C_2, \dots, C_c\}$.

Step1: Build the array $\mathbf{r}^* = \{r_1^*, r_2^*, \dots, r_{c+1}^*\}$, such that each $r_i^* \in \mathbf{r}^*$ is the center of cluster $C_i^* \in \mathbf{C}^*$. Calculate sc value for each C_i^* in \mathbf{C}^* using equation (7), delete the center of cluster with the least sc value from \mathbf{r}^* , recalculate the cluster centers (procedure 3.1.1). Store the new center as $\mathbf{r} = \{r_1, r_2, \dots, r_c\}$.

Step2: Output the new cluster scheme $\mathbf{C} = \{C_1, C_2, \dots, C_c\}$ based on \mathbf{r} .

Procedure 3.1.1 Recalculate the Cluster centers

Input: The array of cluster centers $\mathbf{r}^* = \{r_1^*, r_2^*, \dots, r_c^*\}$ for data set $X = \{x_1, x_2, \dots, x_N\}$.

Output: New array of cluster centers $\mathbf{r} = \{r_1, r_2, \dots, r_c\}$

Step1: Choose the nearest center r_i^* for each element $x_j \in X$, and group x_j into cluster C_i^* whose center is r_i^* .

Step2: Calculate the object median for each C_i^* as the new center for it, denote it as r_i , group all the new centers into array \mathbf{r} , such that $\mathbf{r} = \{r_1, r_2, \dots, r_c\}$.

Step3: if $(\mathbf{r}^* \neq \mathbf{r})$ and maximum step is not reached, go to step1.

Step4: Output \mathbf{r} .

C. Multi-step Maxmin Algorithm

The Multi-step maxmin algorithm, which is another component of 3M algorithm, is used to find an optimal cluster scheme at the first stage of 3M algorithm for a large c value, as well as to find an optimal cluster scheme after each merging process until $c \leq 2$. In multi-step maxmin algorithm, each iteration of optimization process is based on the partitions obtained by maxmin method for a different start point, which always tries to make the cluster as separate as possible. This is why we call it multi-step maxmin. Maxmin

method is originally proposed by J. T. Tou and R. Gonzalez in [13]. We have modified its termination condition such that the algorithm will terminate once a given number of clusters are obtained. Before the presentation of multi-step maxmin algorithm, we give a brief description of the modified maxmin method:

Step 1. Let $X=\{x_1, x_2, \dots, x_N\}$ be a data set. Let x_i be the first cluster center, denoted as r_1 .

Step 2. Determine the farthest object from r_1 and designate it as r_2 . Compute the distance from each remaining object to r_1 and r_2 . For every pair of these distances, we only save the minimum distance. Then select the object having the maximum of these minimum distances as cluster center r_3 .

Step 3. Compute the distance from each of the three objects r_1, r_2, r_3 to the remaining objects and save the minimum of each group of the three distances. Then select the maximum of these minimum distances as the new center again.

Step 4. Repeat the above procedure until enough number of centers is obtained.

Step 5. Assign the remaining objects to its nearest center.

We can see from the above description, most of the cluster centers are distributed around the cluster boundaries, which is not reasonable. Another problem about this method is there is no optimization process. The proposed multi-step maxmin, however, will gradually adjust cluster centers to optimal positions by repeatedly performing the maxmin algorithm and the optimization process.

Algorithm 3.2 Multi-step Maxmin Algorithm

Input: Data set $X = \{x_1, x_2, \dots, x_N\}$,

The number of clusters c ,

Start Point p ,

Integer i .

Output: Cluster scheme $C = \{C_1, C_2, \dots, C_c\}$

Step1. Initialize separation-compactness value SC , $SC = 0$;

Step2. Using p as the start point to perform modified maxmin method to get an cluster scheme $C^* = \{C^*_1, C^*_2, \dots, C^*_c\}$;

Step3. Recalculate the cluster centers (procedure 3.1.1) for C^* .

Step4. Calculate the separation-compactness SC^* value for C^* ; If $SC^* > SC$, $SC = SC^*$, $C = C^*$;

Step5. if $i > c$, $i = 1$; $p = r^*_i$, where r^*_i is the center of C^*_i ; $i = i + 1$; Go to step2 until it converges or a chosen maximum number of steps is reached.

Step6. Output C .

It should be noticed that the multi-step maxmin is somewhat sensitive to the start point. Different start points may lead to different cluster results. However, this problem disappears when the number of clusters is big enough. Fortunately, since at the initial step of 3M algorithm, multi-step maxmin can always be started with a large enough number of clusters as the parameter, sensitivity to the start point is not a concern any more.

C. The Main Algorithm---Multi-step Maxmin and Merging Algorithm (3M Algorithm)

Algorithm 3.1 Multi-step Maxmin and Merge Algorithm (3M algorithm)

Input: Proximity data set $X = \{x_1, x_2, \dots, x_N\}$

$maxnum$ (The maximum number of clusters)

Output: Optimal Cluster Scheme $C = \{C_1, C_2, \dots, C_c\}$

Step1: $c_{opt} = maxnum$; $c = maxnum$; $i = 1$; randomly choose a object $x \in X$ as the start point p ; perform multi-step maxmin algorithm (Algorithm 3.2) based on parameter X, c, i and p to find the optimal cluster scheme $C = \{C_1, C_2, \dots, C_c\}$ for c . Calculate validity function SC for C using equation(3), denote it as SC .

Step2: Perform *merge process* (Algorithm 3.1) to get candidate cluster scheme $C' = \{C'_1, C'_2, \dots, C'_{c-1}\}$; choose the center of C'_1 as start point p ; $c = c - 1$; $i = 2$; perform multi-step maxmin algorithm (Algorithm 3.2) based on parameter X, c, i and p to find the optimal cluster scheme $C^* = \{C^*_1, C^*_2, \dots, C^*_c\}$ for c . Calculate validity function SC for C^* using equation (3), denote it as SC^* ; If $SC^* > SC$, $SC = SC^*$, $C = C^*$, $c_{opt} = c$. Repeat step2 until $c \leq 2$.

Step3: Output $C = \{C_1, C_2, \dots, C_{c_{opt}}\}$ as an optimal cluster scheme.

IV. EXPERIMENTAL RESULTS

In order to test the effectiveness of 3M algorithm in finding an optimal cluster scheme for proximity data, we show some experimental results of the 3M algorithm on both synthetic and real-world data sets and make some comparisons between 3M and other clustering algorithms. For 3M and the alternative algorithms that can be applied to proximity data, all the data sets are first transformed into proximity data, while the ones that cannot still make use of the actual vector representations.

A. Synthetic Data Set

In order to compare 3M algorithm with other clustering methods, we generate a simple synthetic data set with 43 points, which is plotted in figure 1. As being marked in figure 1, the cluster structure of this data set is clear, so that we can easily judge the performance of each method. We can also see that the points are unevenly distributed in this data set — cluster 1 has higher density than cluster 2 and cluster 3, and “S” is obviously an outlier. We perform 3M, k-medoids, subtractive, maxmin, and fuzzy c-means algorithms on this data set. For the first four algorithms, this data set are transformed into proximity data by calculating the Euclidean distance of each pair of points, while the fuzzy c-means still makes use of the actual vector representations. Some results of the five algorithms are shown in figure 2 to figure 6; for 3M algorithm, the validity values calculated by equation (3) for the number of clusters in the range [2, 10] are plotted in figure 7; and detailed comparison of these algorithms on this synthetic data set is available in table 1.

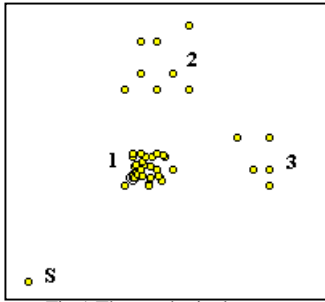


Fig.1 The synthetic data set

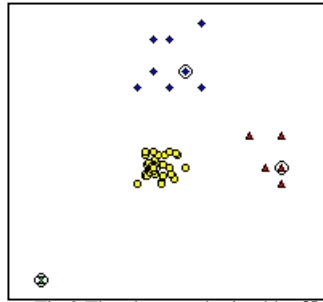


Fig.2 The clusters obtained by **3M**.
(*start point*: any; *maxnum*: any in the range [4, 42])

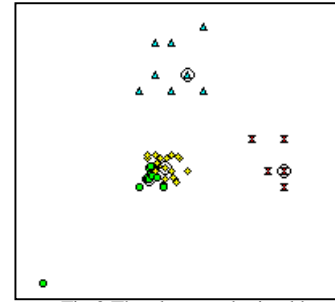


Fig.3 The clusters obtained by **k-medoids** (*number of clusters*: 4)

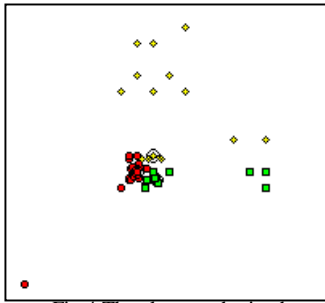


Fig.4 The clusters obtained by **subtractive algorithm**.
(*influence range*: 0.1)

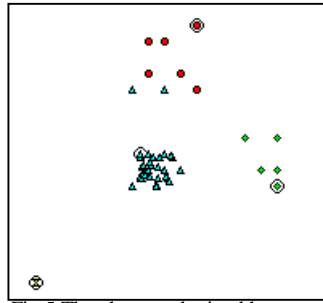


Fig.5 The clusters obtained by **maxmin**
(*threshold coefficient*: 0.5; *start point*: 42)

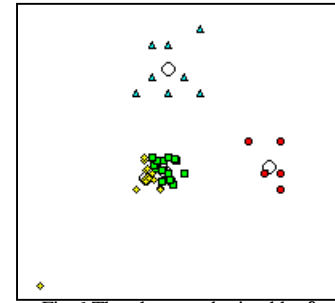


Fig.6 The clusters obtained by **fcm**
(*number of clusters*: 4)

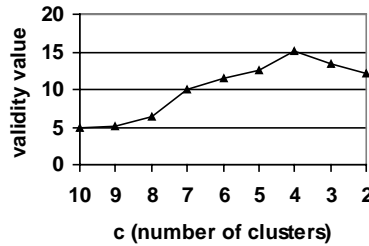


Fig. 7 The validity value for each c value in the range [2,10] on this synthetic data set

TABLE 1. COMPARISONS OF THE FIVE CLUSTERING ALGORITHMS ON THE SYNTHETIC DATA SET

Algorithm	Applicable to proximity data?	Need to specify the number of clusters?	Sensitive to parameters or input order?	Can deal with this unevenly distribution?	Can detect this outlier?
3M	Yes	No	No	Yes	Yes
k-medoids	Yes	Yes	Yes	Yes	No
Subtractive	Yes	No	Yes	No	No
Maxmin	Yes	No	Yes	Yes	Yes
fuzzy c-means	No	Yes	Yes	Yes	No

From this experiment we can see, 3M algorithm is able to find the optimal cluster schemes for proximity data without being given the number of clusters or any other specific parameter, and is stronger in dealing with unevenly distributed data and detecting the outlier than the other alternatives.

B. Wisconsin Breast Cancer Data

Wisconsin breast cancer databases were obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg [16]. This data set contains 699 instances that fall into two classes: benign (458 instances)

and malignant (241 instances). Each instance is represented by 9 attributes, all of which are scaled to a [0, 1] range. As before, we transform the data into proximity data for 3M algorithm by calculating the Euclidian distance between each pair of instances.

In order to test the influence of the two parameters, *start point*, *maxnum* on the performance of 3M algorithm again, we design two sets of experiments. The first set of experiments contains fifteen individual experiments, where we fix the start point to be the first object, and randomly choose a number from the range [20, 200] as the parameter *maxnum* each time. For the second set, which also contains

fifteen individual experiments, we fix the *maxnum* to be twenty, and then randomly choose an object as the start point. For all the tests, we get the same clustering scheme that contains two clusters with the size 233 and 466 objects respectively. It shows again that 3M algorithm is not sensitive to these two parameters. The validity values calculated by equation (3) for the number of clusters in the range [2, 10] are plotted in figure 8.

In order to test whether the cluster scheme obtained by 3M is an optimal one or not, we fix the number of clusters to two for fuzzy c-means and k-medoids algorithms and run them on this data set. K-medoids uses proximity data as input, while fuzzy c-means makes use of the actual vector representations. We randomly choose the initial partitions for these two algorithms and run ten times for each, and then choose the best results for these two. As we can see from the results

shown in table 2, k-medoids obtained the same results as 3M algorithm, and the performance of both of them is a little bit better than the one of fuzzy c-means for this data set, even though object median is less precise than mean vector.

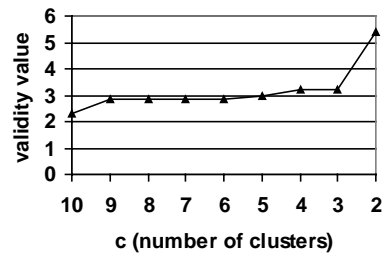


Fig. 8 The validity value for each c value in the range [2,10] on Wisconsin Breast Cancer Data Set

TABLE 2 THE RESULTS OF THE THREE ALGORITHMS ON WISCONSIN BREAST CANCER DATA

Algorithm	Cluster Number	Cluster Center	Error Number	Error Rate (%)
3M	2	212 : (0.6,0.6,0.6,0.5,0.4,1,0.7,0.6,0.2) 57 : (0.3,0.1,0.1,0.1,0.2,0.1,0.2,0.1,0.1)	29	4.15
FCM	(specified) 2	(0.7166,0.6814,0.6762, 0.5775,0.5451,0.7793,0.6107,0.6114,0.2576) (0.3176,0.1472,0.1602,0.1466,0.2206,0.1624,0.2232,0.1400,0.1144)	32	4.58
k-medoids	(specified) 2	212 : (0.6,0.6,0.6,0.5,0.4,1,0.7,0.6,0.2) 57 : (0.3,0.1,0.1,0.1,0.2,0.1,0.2,0.1,0.1)	29	4.15

V. CONCLUSION

Of all the clustering algorithms, only a few can be directly applied to proximity data. These methods require that the number of clusters or some threshold value is given and are always sensitive to some user-selected parameters. In order to remedy these weaknesses, we introduce a new cluster validity function, which works well even when the number of clusters is very large. In addition, a new algorithm, called the multi-step maxmin and merging algorithm (the 3M algorithm) is proposed in this paper. This algorithm extends the basic maxmin method with optimization steps and combines it with a merging strategy such that it can always generate optimal cluster schemes for varying number of clusters. Then the new cluster validity function, which is based on separation and compactness measure, is applied to choose an optimal cluster scheme. Experiments show that the 3M algorithm obtains the best results for the synthetic data compared to three other algorithms applicable to proximity data and fuzzy c-means, and an optimal cluster scheme for Wisconsin Breast Cancer data. Both this two experiments show that 3M is not sensitive to parameters such as the maximum number of clusters chosen to start with and the object used as the start point. More tests and comparisons will be done on both the new validity function and the 3M algorithm in the future.

References

[1] J. MacQueen, "Some methods for classification and analysis of multivariate observations," Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281-297, 1967.

[2] L. Kaufman and P. J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, John Wiley & Sons, New York, 1990.

[3] X. Zhao, "Space Transformation and Clustering Methods for Proximity Data Set," Master Thesis, CACS, University of Louisiana at Lafayette, 2000.

[4] C. Looney, "A Fuzzy Clustering and Fuzzy Merging Algorithm," Technical Report, CS-UNR-101-1999. <http://citeseer.nj.nec.com/looney99fuzzy.html>

[5] U. Kaymak, "A unified approach for practical applications of fuzzy clustering," Proc. 20th Belgium-Netherlands Conference on Artificial Intelligence, BNAIC'00, 2000.

[6] R. R. Yager and D. P. Filev, "Approximate clustering via the mountain method," IEEE Transactions on Systems, Man, and Cybernetics, vol. 24, pp. 1279-1284, 1994.

[7] S. L. Chiu, "Fuzzy model identification based on cluster estimation," Journal of Intelligent and Fuzzy Systems, 2(3), 1994.

[8] J. C. Bezdek, Pattern Recognition with Fuzzy Objective Function, Plenum Press, New York, 1981.

[9] D. E. Gustafson and W. C. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," Proc. IEEE CDC, pp. 761-766, San Diego, USA, 1979.

[10] U. Kaymak and R. Babuška, "Compatible cluster merging for fuzzy modeling," Proc. 4th IEEE International Conference on Fuzzy Systems, vol. 2, pp. 897-904, Yokohama, Japan, March 1995.

[11] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, no. 8, 841-847, 1991.

[12] J. C. Bezdek, "Numerical taxonomy with fuzzy sets," J. Math. Biol., vol. 1, pp. 57-71, 1994.

[13] J. T. Tou and R. Gonzalez, Pattern Recognition Principle, Addison-Wesley, Reading, MA, 1972.

[14] G. H. Ball and D. J. Hall. "ISODATA: an iterative method of multivariate analysis and pattern classification," Proc. IEEE Int. Communications Conf., 1966.

[15] J. Han and M. Kamber, Data Mining: Concepts and Techniques, Academic Press, San Diego, 2001.

[16] O. L. Mangasarian and W. H. Wolberg, "Cancer diagnosis via linear programming," SIAM News, vol. 23, no. 5, September 1990.