Available online at www.sciencedirect.com
SCIENCE DIRECT®
International Journal of Approximate Reasoning
40 (2005) 109–124
INTERNATIONAL JOURNAL OF APPROXIMATE REASONING
www.elsevier.com/locate/ijar

A new fuzzy clustering algorithm for optimally finding granular prototypes

Ying Xie [a,*], Vijay V. Raghavan [a], Praveen Dhatric [a], Xiaoquan Zhao [b]

[a] *The Center for Advanced Computer Studies, University of Louisiana at Lafayette, Lafayette, LA 70504-4330, USA*
[b] *GE Medical Systems, USA*

Received 1 July 2004; accepted 1 November 2004
Available online 7 January 2005

Abstract

Prototype Reasoning using granular objects is an important technology for knowledge discovery. Fuzzy clustering can be used to generate prototypes with different granularities. In order to find optimal granular prototypes through fuzzy clustering, for given data, two conditions are necessary: a good cluster validity function, which can be applied to evaluate the goodness of cluster schemes for varying number of clusters (different granularity); a good cluster algorithm that can produce an optimal solution for a fixed number of clusters. To satisfy the first condition, a new validity measure called granularity–dissimilarity (GD) measure is proposed, which is stable in evaluating granularities and works well even when the number of clusters is very large. For the second condition, we propose a new algorithm called multi-step maxmin and merging algorithm (3M algorithm). Experiments show that, when used in conjunction with the new cluster validity measure, 3M algorithm produces better results on the experimental data sets than several alternatives.

Keywords: Prototype Reasoning; Fuzzy clustering; 3M algorithm; Granular prototype

* Corresponding author. Tel.: +1 337 233 7347; fax: +1 337 482 5791.
E-mail addresses: yxx2098@cacs.louisiana.edu (Y. Xie), raghavan@cacs.louisiana.edu (V.V. Raghavan), pkd5618@cacs.louisiana.edu (P. Dhatric), xiaoquan.zhao@med.ge.com (X. Zhao).

0888-613X/$ - see front matter
doi:10.1016/j.ijar.2004.11.002

1. Introduction

As an extension of instance based reasoning (IBR), prototype based reasoning (PBR) using granular objects [11] offers a powerful paradigm for problem solving. This paradigm can be widely used in the fields such as text and image retrieval, information filtering, scientific data analysis, machine learning and knowledge discovery. Fuzzy modeling technique, where each fuzzy rule can be viewed as a prototype, is a typical example that utilizes PBR paradigm [11]. The first and most important step of successfully applying this paradigm, undoubtedly, is to obtain optimal granular prototypes from the given data. Applying fuzzy clustering is a natural way to obtain granular prototypes. Fuzzy clustering process partitions data into several fuzzy clusters (sets). Each fuzzy cluster can then be utilized as a granular prototype for the purposes of reasoning. (In this paper, the term prototype will be used as a synonym of fuzzy cluster in order to emphasize that the purpose of clustering analysis in this paper is for PBR, rather than the detection of density areas with arbitrary shape.) However, for data set where the user has little prior knowledge, it is quite difficult to find optimal granular prototypes by utilizing the existing fuzzy algorithms such as fuzzy c-means [15], Gustafson–Kessel Algorithm [5], fuzzy isodata [9,12] and subtractive method [3,4]. The reasons are as follows:

(1) The number of clusters, which is one of the most important factors that determine clustering quality and granularity of prototypes, must be given explicitly or implicitly as certain input parameter, such as the expected number of clusters or the minimal density value. Thus, it is difficult to guarantee that the resulting cluster scheme 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 the one used by compatible clustering merging algorithm [6] and extended fuzzy c-means [2]. 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 that the clustering result after a merge is 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 [7], Xie and Beni report that partition coefficient [8] has monotonic decreasing tendency with the number of clusters; through experiments, we find that both Partition Entropy and Average Partition Separability [8] tend to go up as the number of clusters increases. Therefore, those measures are not suitable to be used as the goodness index to determine the number of clusters. Xie and Beni also proposed a compactness-separation validity index S [7], which is considered to be independent of the number of clusters [20]. 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 representative of the ith cluster; x_j is the jth object; and μ_{ij} is the membership value of the jth object belonging to the ith cluster. As one can see, this validity index uses the minimum distance between the cluster representatives as the measure to evaluate the separation of a cluster scheme, which does not seem quite reasonable. Two clustering schemes for a certain data set, even though having the same minimum distance between the representatives 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 [7].

(2) All these methods are sensitive to some initial parameters. For example, fuzzy c-means, just like its crisp counterpart k-means or k-medoids, may give different clustering results with different initial partition. Some density parameter or noise threshold may significantly influence the clustering quality of subtractive clustering as well as other density-based approaches [17]. Thus, even though the number of clusters is given, these methods cannot guarantee that an optimal solution can be obtained.

Therefore, in order to obtain an optimal granular prototype by using fuzzy clustering, two conditions are necessary:

- A good validity function, which can be applied to evaluate of the goodness of cluster schemes for varying number of clusters.
- A good cluster algorithm that can produce an optimal solution for a fixed number of clusters.

Once these two requirements are met, the strategy of getting an optimal fuzzy cluster scheme is straitforward: produce an optimal solution for each potential number of clusters; then use the validity function to choose the best one, so as to automatically decide on the number of clusters.

To satisfy the first condition, we propose a new validity measure called granularity–dissimilarity (GD) measure to evaluate the granularity and dissimilarity of prototypes obtained by fuzzy clustering for a given data set. Experiments show that, unlike the situation when index S is used, our validity measure is more stable in evaluating granularities, and it still works well even when the number of clusters is very large. For the second condition, we propose a new fuzzy clustering algorithm called multi-step maxmin and merging algorithm (3M algorithm). 3M algorithm consists of two components. The first one, which is called multi-step maxmin [19], extends the basic maxmin partition method [16] with a new optimization process. Maxmin method, which tries to make the cluster as separate as possible, provides the initial partition. Then, the added optimization process is performed in order to get the local optimum. By using different cluster representatives as different start points, the maxmin process and the optimization process are repeated until the algorithm converges

or a maximum number of steps is reached. Multi-step maxmin is used to obtain an optimal clustering scheme for a given number c, while 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 an optimal solution for $c-1$. Then the newly proposed validity measure is used to evaluate the goodness of solutions for each c. The cluster scheme with the largest value of validity value is reported as the final optimal solution.

The rest of this paper will be organized as follows. The new fuzzy cluster validity measure is defined in Section 2. Section 3 provides a detailed description of 3M algorithm. In Section 4, complexity analysis of 3M algorithm is conducted. Section 5 reports the experimental results. Finally, Section 6 concludes the paper.

2. A new fuzzy validity measure

Recall that the purpose of fuzzy clustering analysis in this paper is to find optimal granular prototypes each of which can represent a group of homogeneous data. By optimal granularity we mean that, on one hand, each prototype should have fine granularity so that its representing ability is high; on the other hand, the dissimilarity among prototypes should be as large as possible. Therefore our new validity measure is based on measures of both granularity and dissimilarity of the cluster scheme and tries to find a good tradeoff between these two.

Definition 1. Given a cluster scheme $\mathbb{C} = \{C_1, C_2, \ldots, C_c\}$ for a data set $X = \{x_1, x_2, \ldots, x_n\}$, let $\mathbb{C}' = \{C_{pi} \,|\, C_{pi} \in \mathbb{C} \text{ and } C_{pi} \text{ is not singleton}, \ i = 1, 2, \ldots, k$ where $k = |\mathbb{C}'|\}$, the granularity, GR, of the cluster Scheme $\mathbb{C}$ is given by

$$\mathrm{GR} = \frac{\sum_{i=1}^{k} \dfrac{\sum_{x_j \in C_{pi}, x_j \neq r_i} \mu_i(x_j)^2 d(x_j, r_i)^2}{\sum_{x_j \in C_{pi}, x_j \neq r_i} \mu_i(x_j)^2}}{k}, \tag{1}$$

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

Definition 2. The dissimilarity, DS, of a cluster scheme $\mathbb{C} = \{C_1, C_2, \ldots, C_c\}$ for a data set $X = \{x_1, x_2, \ldots, x_n\}$ is given by

$$\mathrm{DS} = \left(\frac{\sum_{i=1}^{c} \min_{1 \leqslant j \leqslant c, i \neq j} \{d(r_i, r_j)\}}{c} \right)^2, \tag{2}$$

where c is the number of clusters, r_i is the representative of ith cluster, and $d(r_i, r_j)$ is the distance between r_i and r_j.

Definition 3. Given a cluster scheme $\mathbb{C} = \{C_1, C_2, \ldots, C_c\}$ for a data set $X = \{x_1, x_2, \ldots, x_n\}$, let $\mathbb{C}' = \{C_{pi} | C_{pi} \in \mathbb{C}$ and C_{pi} is not singleton, $i = 1, 2, \ldots, k$ where $k = |\mathbb{C}'|\}$. The granularity–dissimilarity, GD, of the cluster scheme C is given by

$$\mathrm{GD} = \frac{k}{c} \times \frac{\mathrm{DS}}{\mathrm{GR}}. \tag{3}$$

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

$$\max_{2 \leqslant c < n} \left\{ \max_{\Omega_c} \{\mathrm{GD}\} \right\}, \tag{4}$$

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

3. 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 optimal partitions for the largest possible c value at the initial stage, as well as for c value after each merging process until $c \leqslant 2$. 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 fuzzy membership function that are suitable for both data in vector representations as well as proximity data, where just pairwise distance are given.

3.1. Object median and fuzzy membership function

There are scenarios, such as in some forms of text and image retrieval, where we need employ object median [14] instead of mean to represent each prototype. In order to make our algorithm fit all kinds of situations, in this paper, we utilize object median as the representative of each cluster. Let C_i be a subset of a data set with distance function d, a point x_0 in C_i is called a object median of C_i if

$$\sum_{y \in C_i} d(x_0, y) = \min_{x \in C_i} \left\{ \sum_{y \in C_i} d(x, y) \right\}. \tag{5}$$

If the vector representation of each object is available, an alternative definition of object median can be given as follows: a point x_0 in C_i is called a object median of C_i if

$$d(x_0, m_i) = \min_{x \in C_i}(x, m_i), \tag{6}$$

where m_i is the mean of cluster C_i. The advantage of formula (6) is that it has linear time complexity in terms of the number of objects in the cluster, while formula (5) has quadratic complexity. Let $X = \{x_1, x_2, \ldots, x_n\}$ be a data set, r_j be the representative of the jth cluster, $j = 1, 2, \ldots, c$. We define membership functions μ_{C_j}, $j = 1, 2, \ldots, c$, for any $x \in X$, as

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

It is easy to see that

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

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

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

3.2. Merging algorithm

The merging process generally used by earlier studies involves some similarity or compatibility measure to choose the most similar or compatible pair of clusters for merging [1,6,13]. In our merging process, however, we choose the "worst" cluster and delete it. Each object included in this cluster will be placed into its own nearest cluster. Then, all involved clusters will be adjusted. As one can see, our merging process may affect multiple clusters, which we consider to be more practical. How to choose the "worst" cluster? We still use the measures of granularity and dissimilarity to evaluate each individual cluster (except singleton).

Definition 4. Given a cluster scheme $\mathbb{C} = \{C_1, C_2, \ldots, C_c\}$ for a data set $X = \{x_1, x_2, \ldots, x_n\}$, for each $C_i \in \mathbb{C}$, if C_i is not a singleton, the granularity of C_i, denoted as gr_i, is given by

$$gr_i = \frac{\sum_{x_j \in C_i, x_j \neq r_i} \mu_i(x_j)^2 d(x_j, r_i)^2}{\sum_{x_j \in C_i, x_j \neq r_i} \mu_i(x_j)^2}, \tag{7}$$

where $\mu_i(x_j)$ is the membership value of x_j belonging to ith cluster C_i, r_i is the representative of the ith cluster C_i, c is the number of clusters, and $2 \leqslant c < N$. if C_i is a singleton, its granularity is defined to be 0.

Definition 5. Given a cluster scheme $\mathbb{C} = \{C_1, C_2, \ldots, C_N\}$ for a data set $X = \{x_1, x_2, \ldots, x_n\}$, for each $C_i \in \mathbb{C}$, the dissimilarity of C_i, denoted as ds_i, is given by

$$\mathrm{ds}_i = \left(\min_{1 \leqslant j \leqslant c, i \neq j} \{d(r_i, r_j)\} \right)^2, \tag{8}$$

where r_i is the representative of the ith cluster C_i, r_j is the representative of the jth cluster C_j, c is the number of clusters, and $2 \leqslant c < N$.

Then, based on the above definitions, we first choose the clusters with the least dissimilarity value, from which we select the one with largest granularity value as the "worst" cluster. If there are ties on the largest granularity value, we randomly choose one from the tie as the "worst" cluster.

Algorithm 1. Merging algorithm

Input: Optimal cluster scheme $\mathbb{C}^* = \{C_1^*, C_2^*, \ldots, C_{c+1}^*\}$ for a data set $X = \{x_1, x_2, \ldots, x_n\}$ where $c \geqslant 2$.

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

Step 1: Build the array $R^* = \{r_1^*, r_2^*, \ldots, r_{c+1}^*\}$, such that each $r_i^* \in R^*$ is the representative of cluster $C_i^* \in \mathbb{C}^*$. Choose the "worst cluster" from $\mathbb{C}^*$ and delete its representative from R^*, recalculate the cluster representatives (Procedure 1.1). Store the new representative as $R = \{r_1, r_2, \ldots, r_c\}$.

Step 2: Output the new cluster scheme $\mathbb{C} = \{C_1, C_2, \ldots, C_c\}$ based on R.

Procedure 1.1. Recalculate the cluster representatives

Input: The array of cluster representatives $R^* = \{r_1^*, r_2^*, \ldots, r_c^*\}$ for data set $X = \{x_1, x_2, \ldots, x_n\}$.

Output: New array of cluster representatives $R = \{r_1, r_2, \ldots, r_c\}$.

Step 1: Choose the nearest representative r_i^* for each element $x_j \in X$, and group x_j into cluster C_i^* whose representative is r_i^*.

Step 2: Calculate the object median for each C_i^* as the new representative for it, denote it as r_i, group all the new representatives into array R, such that $R = \{r_1, r_2, \ldots, r_c\}$.

Step 3: if $R^* \neq R$ and maximum number of loops is not reached, $R^* \leftarrow R$, and go to Step 1.

Step 4: Output R.

3.3. 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 $c \leqslant 2$. In multi-step maxmin algorithm, each iteration of optimization process is

based on the partitions obtained by maxmin method for a different start point. This is why we call it multi-step maxmin. Maxmin method is originally proposed by Tou and Gonzalez in [16]. This method tries to make clusters as separate as possible. We have modified its termination condition so 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 $X = x_1, x_2, \ldots, x_n$ be a data set. Let x_i be the first cluster representative, 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 representative r_3.
Step 3: Compute the distance from each of the remaining object to the three objects r_1, r_2, r_3, and save the minimum of these three distances for that object. Then select the object having the maximum of these minimum distances as the new representative again.
Step 4: Repeat the above procedure until enough number of representatives is obtained.
Step 5: Assign the remaining objects to its nearest representative.

As can be seen from the above description, most of the cluster representatives are distributed around the cluster boundaries, which is not quite reasonable. The vproposed multi-step maxmin will gradually adjust cluster representatives to optimal positions by repeatedly performing the maxmin algorithm and an optimization process.

Algorithm 2. Multi-step maximum algorithm
Input: Data set $X = \{x_1, x_2, \ldots, x_n\}$, the number of clusters c, start Point p, $i \leftarrow 1$.
Output: Cluster Scheme $\mathbb{C} = \{C_1, C_2, \ldots, C_c\}$.
Step 1: Initialize granularity–dissimilarity value GD to be zero.
Step 2: Using p as the start point to perform modified maxmin method to get an cluster scheme $\mathbb{C}^* = \{C_1^*, C_2^*, \ldots, C_c^*\}$.
Step 3: Recalculate the cluster representatives (Procedure 1.1) for $\mathbb{C}^*$.
Step 4: Calculate the granularity–dissimilarity value GD^* value for $\mathbb{C}^*$; If $GD^* > GD$, $GD = GD^*$, $\mathbb{C} = \mathbb{C}^*$.
Step 5: $i \leftarrow i + 1$; $p \leftarrow r_i^*$, where r_i^* is the representative of C_i^*; Go to Step 2 until $i > c$, where c is the number of clusters.
Step 6: Output $\mathbb{C}$.

3.4. The main algorithm—Multi-step maxmin and merging algorithm (3M algorithm)

Algorithm 3. Multi-step maxmin and merge algorithm (3M algorithm)

Input: Data set $X = \{x_1, x_2, \ldots, x_n\}$, *maxnum* (the maximum number of clusters).

Output: Optimal cluster scheme $\mathbb{C} = \{C_1, C_2, \ldots, C_c\}$.

Step 1: $c_{\text{opt}} \leftarrow maxnum$; $c \leftarrow maxnum$; $i \leftarrow 1$; randomly choose a object $x \in X$ as the start point p; perform multi-step maxmin algorithm (Algorithm 2) based on parameter X, c, i and p to find the optimal cluster scheme $\mathbb{C} = \{C_1, C_2, \ldots, C_c\}$ for c. Calculate validity function GD for $\mathbb{C}$ using Eq. (3), denote it as GD.

Step 2: Perform merging process (Algorithm 1) to get candidate cluster scheme $\mathbb{C}' = \{C'_1, C'_2, \ldots, C'_{c-1}\}$; choose the representative of C'_1 as start point p; $c \leftarrow c - 1$; $i \leftarrow 2$; perform multi-step maxmin algorithm (Algorithm 2) based on parameter X, c, i and p to find the optimal cluster scheme $\mathbb{C}^* = \{C^*_1, C^*_2, \ldots, C^*_c\}$ for c. Calculate validity function GD for C^* using Eq. (3), denote it as GD^*; If $\text{GD}^* > \text{GD}$, $\text{GD} \leftarrow \text{GD}^*$, $C \leftarrow C^*$, $c_{\text{opt}} \leftarrow c$. Repeat step2 until $c \leqslant 2$.

Step 3: Output $\mathbb{C} = \{C_1, C_2, \ldots, C_{\text{opt}}\}$ as an optimal cluster scheme.

4. Complexity analysis

In this section, we will analyze the time complexity of 3M algorithm. Assume that the data set has n objects, and the maximum number of clusters specified by the user is c. First we run maxmin algorithm to obtain an initial partition, whose time complexity is $O(c \cdot n)$. Then we adjust the representative of each cluster to its object median according to formula (6). The time complexity of this step is $O(m \cdot c \cdot n)$, where m is either the number of loops reaching convergence or the maximum number of loops specified by the user. The calculation of fuzzy membership value for each point and the calculation of validity value are both $O(c \cdot n)$. Up to now, the time complexity is $O(m \cdot c \cdot n)$. The above process needs to be conducted for multiple c times, after which the time complexity will be $O(m \cdot c^2 \cdot n)$.

The complexity of merging process is the same as the process of adjusting clustering representatives, which is $O(m \cdot c \cdot n)$. Therefore, the total time complexity of 3M algorithm is $\sum_{i=2}^{c} O(m \cdot i^2 \cdot n) = O(m \cdot c^3 \cdot n)$. Since the process of adjusting the representative of each cluster to its object median always begins with a good initial partition (compared with random initial partition used by fuzzy c-means, k-medoids), either generated by maxmin algorithm or merging process, this optimization process is anticipated to be able to converge after a relatively small number of loops (i.e., m is always a very small integer. For all the experiments we conducted, $m < 5$). Even if it is not the case, m at most reaches the number specified by the user. Therefore, m can be omitted as a constant, which makes the time complexity to be $O(c^3 \cdot n)$.

As one can see, c, the maximum expected number of clusters that is provided by the user, plays a great role in computing time of 3M algorithm. In many cases, c^3 can be viewed as an acceptable constant, which makes 3M algorithm linear. However, if

the size of the data set is very large and the maximum expected number of clusters is very big (in other words, if $c^3 \cdot n$ is too big to be processed using available computer resources), the following strategies can be applied to reduce the time complexity:

- If the user has prior knowledge that the possible number of clusters is falling into a small range, the number of clusters beyond this range can be completely ignored, which makes the time complexity reduce to $O(c^2 \cdot n)$.
- If assuming that the candidate cluster scheme generated by merging process as the optimal one for the corresponding number of clusters, we can ignore the following multi-step maxmin process, which reduces the time complexity to $O(c^2 \cdot n)$.
- If c^3 is extremely big, we can first begin with a very small number of clusters (e.g. $c = 3$) and run multi-step maxmin clustering algorithm to obtain a small set of partitions. This process can be repeated hierarchically until each partition obtained can be handled by 3M algorithm.

For applications where proximity data are used (vector representation is not available), we have to use formula (5) to obtain object median as the representative for each cluster, which makes the complexity to be about $O(c \cdot n^2 + c^3 \cdot n)$. The n^2 term is coming from the calculation of formula (5), which is the price we have to pay for clustering proximity data, no matter which clustering algorithm is used.

5. Experimental results

In this section, we will focus on testing the effectiveness of 3M algorithm in finding optimal cluster scheme for given data. We utilize four different data sets whose cluster structures have already been known in advance. Therefore, our experiments will examine whether 3M algorithm plus GD measure is able to automatically detect those natural cluster structures or not. Based on the experimental data, we also make comparisons both between 3M and several other clustering algorithms and between GD measure and Xie–Beni measure. The final experimental results show that 3M algorithm plus GD measure is able to produce optimal cluster schemes for all the data sets we use and its performance is better than the ones provided by several alternatives.

5.1. Synthetic data

In order to compare 3M algorithm with other clustering methods, we generate a simple synthetic data set with 43 points, which is plotted in Fig. 1. As being marked, the cluster structure of this data set is clear, so that we can easily judge the performance of each clustering method. One can also see that 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. Some results of the five algorithms are shown in Fig. 1; for 3M algorithm, the validity values calculated by Eq. (3) for the number of

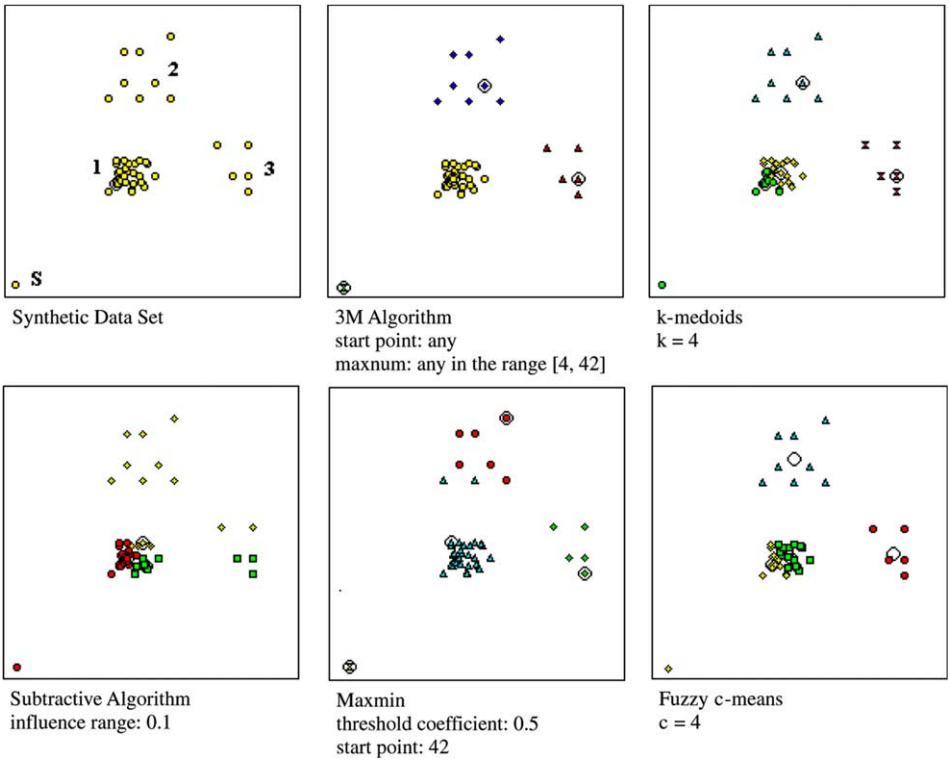

Fig. 1. The experimental results on synthetic data.

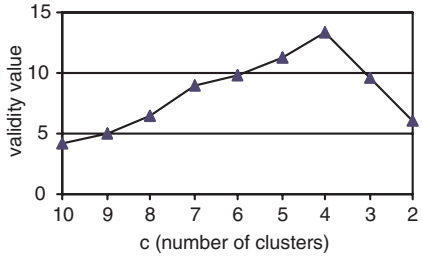

Fig. 2. 3M algorithm: "validity/number of clusters" curve for synthetic data.

clusters in the range [2, 10] are plotted in Fig. 2; and detailed comparison of these algorithms on this synthetic data set is available in Table 1.

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

Table 1
Comparisons of the five clustering algorithms on synthetic data

Algorithm	Need to specify the number of clusters?	Sensitive to parameters or input order?	Can handle this unevenly distribution?	Can detect this outlier?
3M	No	No	Yes	Yes
k-Medoids	Yes	Yes	Yes	No
Subtractive	No	Yes	No	No
Maxmin	No	Yes	Yes	Yes
Fuzzy c-means	Yes	Yes	Yes	No

that the strategy of using validity measure plus fuzzy c-means to generate optimal cluster scheme adopted in [11] may not always dependable, even when the validity measure is perfect. The reason is for each given c, as this experiment shows, fuzzy c-means may not even get the optimal cluster scheme.

5.2. Wisconsin breast cancer data

Wisconsin breast cancer databases were obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg [10]. This data set contains 699 instances that fall into two classes: benign (458 instances) and malignant (241 instances). Each instance is represented by nine attributes, all of which are scaled to a [0, 1] range.

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 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 Eq. (3) for the number of clusters in the range [2, 10] are plotted in Fig. 3.

Now, we fix the number of clusters to two and run both fuzzy c-means and k-medoids algorithms, As we can see from the results shown in Table 2, k-medoids ob-

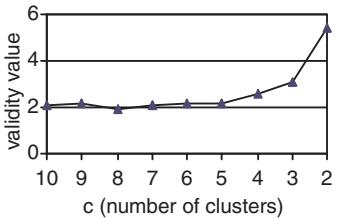

Fig. 3. 3M algorithm: "validity/number of clusters" curve for Wisconsin breast cancer data.

Table 2
The experimental results on Wisconsin breast cancer data

Algorithm	Clusters number	Cluster representative	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	(0.717, 0.681, 0.676, 0.578, 0.545, 0.779, 0.611, 0.611, 0.258); (0.318, 0.147, 0.16, 0.147, 0.22, 0.162, 0.2223, 0.14, 0.114)	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

tained 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.

5.3. Star data

In order to test the effectiveness of our new validity function GD in detecting the optimal cluster scheme, we conduct both 3M+GD and 3M+Xie–Beni on the data set of 51 bright stars near Polaris [7,12], which is shown in Fig. 4(1). For the convenience of comparison, we take the reciprocal form of Xie–Beni validity measure, so that larger the Xie–Beni validity value, better the cluster scheme by this measure. The optimal cluster number of this data set is 8 or 9 [7]. The experimental results are shown in Fig. 4. As we can see, 3M+GD successfully detect the best two clustering schemes that are shown in Fig. 4(2) and (3) respectively. Xie–Beni measure favors a larger number of clusters. When we limit the number of clusters below 43, 3M+Xie–Beni report that 6 and 11 are two best choices. In Fig. 5, we plot the normalized validity values according to these two measures along number of clusters, from which it can be seen that GD is more stable than Xie–Beni in the sense the number of clusters around the optimal one are also getting higher values than others. This feature is obviously important for the task of finding optimal granular prototypes. Also in this

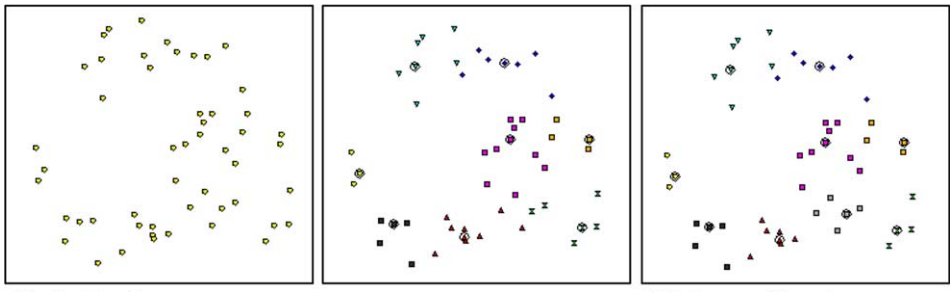

Fig. 4. The experimental results on star data.

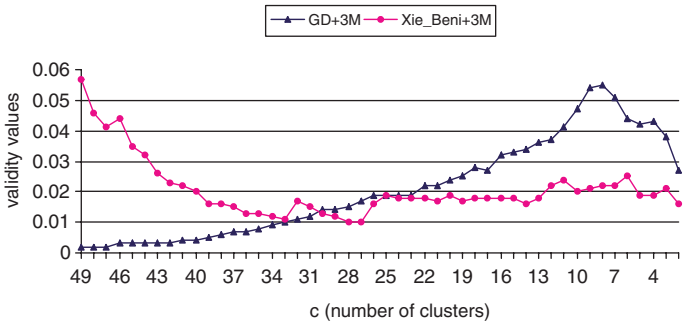

Fig. 5. GD vs. Xie–Beni: "validity/number of clusters" curve for star data.

experiment, the performance of 3M algorithm is not affected by the choice of parameter settings.

5.4. Image data

In this subsection, experiment on 3M algorithm is performed based on image data (Fig. 6). The dataset consists of 75 Images that were selected from Multimedia Information Technology Research Group repository maintained at Pennsylvania State University [21]. The images were selected such that there are seven classes of images based on the color content. The seven classes include people (15), elephants (13), red flowers (14), yellow flowers (9), white flowers (7), mountains (7), and horses in grasslands (10). In this experiment, we utilize the image data as proximity data and the distance matrix of the images are obtained through a technique called color histogram matching [18]. Therefore, for this data set, object median calculated from formula (5) will be used as representative for each cluster. On this image data set, we conduct 3M algorithm with both GD measure and Xie–Beni measure. By

Fig. 6. The experimental results on image data.

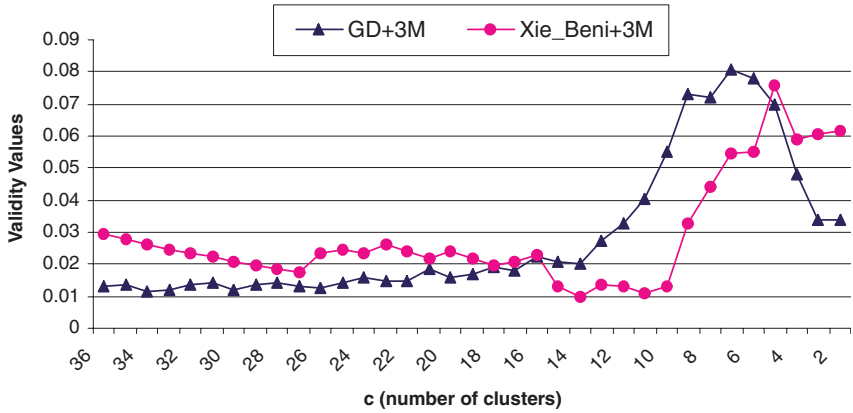

Fig. 7. GD vs. Xie–Beni: "validity/number of clusters" curve for image data.

3M + GD, we get the optimal cluster number 7, while by 3M + Xie–Beni, the reported optimal number of clusters is 5. The normalized validity values along the number of clusters in the range [2, 36] are plotted in Fig. 7 (the Xie–Beni measure is taken in the reciprocal form). And this experiment shows again that GD measure is more stable in evaluating granularity of cluster scheme.

6. Conclusions and future work

Fuzzy clustering is an effective way to find granular prototypes. However, existing fuzzy clustering algorithms generally 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 measure (GD) that is stable in evaluating granularities and works well even when the number of clusters is very large. In addition, in this paper, a new fuzzy clustering algorithm, called the multi-step maxmin and merging algorithm (3M algorithm), is proposed. 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 measure, which is based on granularity and dissimilarity measures, is applied to choose an optimal cluster scheme. Experiments show that 3M algorithm plus GD measure obtains optimal cluster schemes for the synthetic data, Wisconsin Breast Cancer data, star data and the image data. All experiments show that 3M algorithm is not sensitive to parameters such as the maximum number of clusters chosen to start with and the object used as the start point. Our future work will focus on scaling 3M algorithm to large-size data sets. The effectiveness and efficiency of hierarchically utilizing 3M algorithm, which is mentioned in Section 4, will be tested on large-size data set.

References

[1] M. Setnes, U. Kaymak, Extended fuzzy c-means with volume prototypes and cluster merging, in: Proceedings of EUFIT'98, Aachen, Germany, 1998, pp. 1360–1364.
[2] U. Kaymak, A unified approach for practical applications of fuzzy clustering, in: Proceedings of 20th Belgium–Netherlands Conference on Artificial Intelligence, 2000.
[3] R.R. Yager, D.P. Filev, Approximate clustering via the mountain method, IEEE Trans. Systems Man Cybernet. 24 (8) (1994) 1279–1284.
[4] S.L. Chiu, Fuzzy model identification based on cluster estimation, J. Intell. Fuzzy Syst. 2 (3) (1994).
[5] D.E. Gustafson, W.C. Kessel, Fuzzy clustering with a fuzzy covariance matrix, in: Proceedings of IEEE CDC, San Diego, USA, 1979, pp. 761–766.
[6] U. Kaymak, R. Babuka, Compatible cluster merging for fuzzy modeling, in: Proceedings of 4th IEEE International Conference on Fuzzy Systems 2, Yokohama, Japan, 1995, pp. 897–904.
[7] X.L. Xie, G. Beni, A validity measure for fuzzy clustering, IEEE Trans. Pattern Anal. Mach. Intell. 13 (8) (1991) 841–847.
[8] J.C. Bezdek, Numerical taxonomy with fuzzy sets, J. Math. Biol. 1 (1) (1974) 57–71.
[9] G.H. Ball, D.J. Hall, ISODATA: an iterative method of multivariate analysis and pattern classification, in: Proceedings of IEEE International Communications Conference, 1966.
[10] O.L. Mangasarian, W.H. Wolberg, Cancer diagnosis via linear programming, SIAM News 23 (5) (1990).
[11] R.R. Yager, Prototype reasoning and knowledge creation using granular objects, in: Proceedings of IEEE International Conference on Fuzzy Systems, 2002.
[12] M.P. Windham, Cluster validity for fuzzy c-means clustering algorithm, IEEE Trans. Pattern Anal. Mach. Intell. 4 (4) (1982).
[13] C.G. Looney, Interactive clustering and merging with a new fuzzy expected value, Pattern Recognition 35 (11) (2002) 2413–2423.
[14] L. Kaufman, P.J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, John Wiley, New York, 1990.
[15] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function, Plenum Press, New York, 1981.
[16] J.T. Tou, R. Gonzalez, Pattern Recognition Principle, Addison-Wesley, Reading, MA, 1972.
[17] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Academic Press, San Diego, 2001.
[18] D.H. Ballard, C.M. Brown, Computer Vision, Prentice-Hall, Englewood Cliffs, NJ, 1982.
[19] X. Zhao, Space transformation and clustering methods for proximity data set, Master Thesis, CACS, University of Louisiana at Lafayette, 2000.
[20] G.C. Looney, A Fuzzy Clustering and fuzzy merging algorithm, Technical Report CS-UNR-101-1999.
[21] <http://wang.ist.psu.edu/IMAGE>, 2003.