Fuzzy Clustering With Volume Prototypes and Adaptive Cluster Merging

Uzay Kaymak and Magne Setnes

***Abstract*—Two extensions to the objective function-based fuzzy clustering are proposed. First, the (point) prototypes are extended to hypervolumes, whose size can be fixed or can be determined automatically from the data being clustered. It is shown that clustering with hypervolume prototypes can be formulated as the minimization of an objective function. Second, a heuristic cluster merging step is introduced where the similarity among the clusters is assessed during optimization. Starting with an overestimation of the number of clusters in the data, similar clusters are merged in order to obtain a suitable partitioning. An adaptive threshold for merging is proposed. The extensions proposed are applied to Gustafson–Kessel and fuzzy c-means algorithms, and the resulting extended algorithm is given. The properties of the new algorithm are illustrated by various examples.**

***Index Terms*—Cluster merging, fuzzy clustering, similarity, volume prototypes.**

I. Introduction

OBJECTIVE function-based fuzzy clustering algorithms such as the fuzzy c-means (FCM) algorithm have been used extensively for different tasks like pattern recognition, data analysis, image processing and fuzzy modeling. Fuzzy clustering algorithms partition the data set into overlapping groups such that the clusters describe an underlying structure within the data [1]. In order to obtain a good performance from a fuzzy clustering algorithm, a number of issues must be considered. These concern the shape and the volume of the clusters, the initialization of the clustering algorithm, the distribution of the data patterns, and the number of clusters in the data.

In algorithms with point prototypes, the shape of the clusters is determined by the distance measure that is used. The FCM algorithm, for instance, uses the Euclidian distance measure and is thus suitable for clusters with a spherical shape [2]. If *a priori* information is available regarding the cluster shape, the distance metric can be modified to the cluster shape. Alternatively, one can also adapt the distance metric to the data as done in the Gustafson–Kessel (GK) clustering algorithm [3]. Another way to influence the shape of the clusters is to select prototypes with a geometric structure. For example, fuzzy c-varieties (FCV) algorithm uses linear subspaces of the clustering space as prototypes [4], which is useful for detecting lines and other linear structures in the data. The membership of data points to a cluster decreases with their distance to the cluster prototype. In many applications, however, the points close to a cluster prototype can be considered to belong fully to the fuzzy set represented by the cluster. This suggests that the cluster prototypes should extend a certain distance from the cluster centers, so that the data points within these regions belong to the corresponding clusters with membership 1.0. Current clustering algorithms do not have this property.

It is well known that the fuzzy clustering algorithms are sensitive to the initialization. Often, the algorithms are initialized randomly multiple times, in the hope that one of the initializations leads to good clustering results. The sensitivity to initialization becomes acute when the distribution of the data patterns shows a large variance. When there are clusters with varying data density and with different volumes, a bad initialization can easily lead to suboptimal clustering results. Moreover, the intuitively correct clustering results need not even correspond to a minimum of the objective function under these circumstances [5].

One might argue that by carefully guiding the data collection process, one may attempt to obtain roughly the same data density in all interesting regions. Often, however, the analyst does not have control over the data collection process. For example, if the application area is automatic understanding of outdoor scenes, the number of pixels corresponding to different groups (e.g., sky, foliage, ground, etc.) differs from picture to picture and cannot be controlled explicitly. Similarly, dynamic systems may generate more data in certain regions of the state space than others. Hence, a clustering algorithm that is less sensitive to differences in initialization and the distribution of data is desired.

Perhaps the most important parameter that has to be selected in fuzzy clustering is the number of clusters in the data. Objective function-based fuzzy clustering algorithms partition the data into a specified number of clusters, no matter whether the clusters are meaningful or not. The number of clusters should ideally correspond to the number of sub-structures naturally present in the data. Many methods have been proposed to determine the relevant number of clusters in a clustering problem. Typically external cluster validity measures are used [6], [7] to assess the validity of a given partition by considering criteria like the compactness of the clusters and the distance between them. Another approach to determine the number of clusters is using cluster merging, where the clustering starts with a large number of clusters and the compatible clusters are iteratively merged until the correct number of clusters are determined [8], [9]. In addition to the merging, it is also possible to remove unimportant clusters in a supervised fashion [10].

Manuscript received December 5, 2001; revised March 22, 2002 and May 15, 2002.

U. Kaymak is with Erasmus University Rotterdam, Faculty of Economics, Department of Computer Science, 3000 DR Rotterdam, The Netherlands (e-mail: u.kaymak@ieee.org).

M. Setnes is with Control Systems Engineering, Faculty of Information and Technology, Delft University of Technology, NL-2600 GA, Delft, The Netherlands (e-mail: magne@ieee.org).

Digital Object Identifier 10.1109/TFUZZ.2002.805901

In this paper, we propose an extension of objective function-based fuzzy clustering algorithms with volume prototypes and similarity based cluster merging. The goal of this extension is to address the issues discussed in the previous paragraphs. Extended versions of the fuzzy c-means (E-FCM) and the Gustafson–Kessel (E-GK) algorithms are given and their properties are studied. Real-world applications of extended clustering algorithms are not considered in this paper, but the interested reader is referred to [11] for successful application of the E-FCM algorithm in direct marketing.

The outline of the paper is as follows. Section II provides the general formulation of the extended fuzzy clustering proposed. The objective function that clustering with volume prototypes minimizes is presented. The update equations for the alternating optimization are derived, after which a heuristic similarity based cluster merging step is introduced. The algorithm for the extended versions of the Gustafson–Kessel clustering and the FCM clustering is described in Section III. Section IV provides examples that illustrate the properties of the extended algorithm. Finally, conclusions are given in Section V.

II. Extended Fuzzy Clustering

In this section, the extension of fuzzy clustering algorithms with volume prototypes and similarity based cluster merging is described. Let $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ be a set of N data objects represented by n-dimensional feature vectors $\mathbf{x_k} = [x_{1k}, \ldots, x_{nk}]^T \in \mathbb{R}^n$. A set of N feature vectors can then be represented as a $n \times N$ data matrix $\mathbf{X}$. A fuzzy clustering algorithm with point prototypes partitions the data $\mathbf{X}$ into M fuzzy clusters, forming a fuzzy partition in $\mathbf{X}$ based on the distance between the data points $\mathbf{x}_k$ and the cluster prototypes $\mathbf{v}_i \in \mathbb{R}^n$, $i = 1, 2, \ldots, M$ [4]. A fuzzy partition can be conveniently represented as a matrix $\mathbf{U}$, whose elements $u_{ik} \in [0, 1]$ represent the membership degree of $\mathbf{x}_k$ in cluster i.

The general form of the distance measure used is given by

$$d^2(\mathbf{x}_k, \mathbf{v}_i) = (\mathbf{x}_k - \mathbf{v}_i)^T \mathbf{A}_i (\mathbf{x}_k - \mathbf{v}_i) \tag{1}$$

where the norm matrix $\mathbf{A}_i$ is a positive–definite symmetric matrix. The FCM algorithm uses the Euclidian distance measure, i.e., $\mathbf{A}_i = \mathbf{I}\ \forall i$, while the GK algorithm uses the Mahalanobis distance, i.e., $\mathbf{A}_i = \mathbf{P}_i^{-1}$ with the additional volume constraint $|\mathbf{A}_i| = \rho_i$, where $\mathbf{P}_i$ is the covariance matrix of cluster i.

A. Clustering With Volume Prototypes

Volume prototypes extend the cluster prototypes from points to regions in the clustering space. Volume prototypes are quite useful when generating fuzzy rules using fuzzy clustering, since the cores of the fuzzy sets in the rules need not be a single point, allowing the shape of the fuzzy sets to be determined by data rather than the properties of the selected clustering algorithm. The relation of the cluster volumes to the performance of the clustering algorithm has been recognized for a long time. Many cluster validity measures proposed are related to cluster volumes [6], [7]. Other authors have proposed adapting the volume of clusters [12]. Recently, a fuzzy clustering algorithm based on the minimization of the total cluster volume has also been proposed [5].

Often, a number of data points close to a cluster center can be considered to belong fully to the cluster. This is especially the case when there are some clusters that are well separated from the others. It is then sensible to extend the core of a cluster from a single point to a region in the space. One then obtains *volume prototypes* defined as follows.

Definition: A volume prototype $\tilde{\mathbf{v}} \in \mathbb{R}^n$ is a n-dimensional convex and compact subspace of the clustering space.

Note that the volume prototype can have an arbitrary shape and size according to this definition. When the original cluster prototypes are points, it is straightforward to select the volume prototypes $\tilde{\mathbf{v}}_i$ such that they extend a given distance r_i in all directions from the cluster center $\mathbf{v}_i$.

The extended clustering algorithm measures the distance from the data points to the volume prototypes. The data points $\mathbf{x}_k$ that satisfy $d(\mathbf{x}_k, \mathbf{v}_i) \leq r_i$ are elements of the volume prototype $\tilde{\mathbf{v}}_i$ and have by definition maximal membership to that particular cluster. The size of the volume prototypes are thus determined by the radius r_i. With knowledge of the data, this radius can be defined by the user (fixed size prototypes), or it can be estimated from the data. The latter approach is followed.

A natural way to determine the radii r_i, $i = 1, \ldots, M$ is to relate them to the size of the clusters. This can be achieved by considering the fuzzy cluster covariance matrix

$$\mathbf{P}_i = \frac{\sum_{k=1}^{N} u_{ik}^m (\mathbf{x}_k - \mathbf{v}_i)(\mathbf{x}_k - \mathbf{v}_i)^T}{\sum_{k=1}^{N} u_{ik}^m}. \tag{2}$$

The determinant $|\mathbf{P}_i|$ of the cluster covariance matrix gives the volume of the cluster. Because $\mathbf{P}_i$ is a positive–definite and symmetric matrix, it can be decomposed such that $\mathbf{P}_i = \mathbf{Q}_i \mathbf{\Lambda}_i \mathbf{Q}_i^T$, where $\mathbf{Q}_i$ is orthonormal and $\mathbf{\Lambda}_i$ is diagonal with nonzero elements $\lambda_{i1}, \ldots, \lambda_{in}$. We let the volume prototypes extend a distance of $\sqrt{\lambda_{ij}}$, $j = 1, 2, \ldots, n$ along each eigenvector q_{ij}. In the one-dimensional case, this choice implies that the cluster prototype extends one (fuzzy) standard deviation from the cluster center. We make this choice since the points within one standard deviation can be considered not to differ significantly from the cluster center. In the multidimensional case, the size of the radius in each direction is determined by measuring the distances along the transformed coordinates according to

$$\sqrt{\mathbf{\Lambda}_i} \mathbf{Q}_i^T \mathbf{A}_i \mathbf{Q}_i \sqrt{\mathbf{\Lambda}_i} \tag{3}$$

where $\sqrt{\mathbf{\Lambda}_i}$ represents a matrix whose elements are equal to the square root of the elements of $\mathbf{\Lambda}_i$.

When $\mathbf{A}_i$ induces a different norm than given by the covariance matrix, n different values will be obtained for the radius. In that case, a single value can be determined by averaging, as discussed in Section III. The shape of the volume prototypes is the same as the shape of the clusters induced by the distance metric. When Euclidian distance measure is used as in the FCM algorithm, the volume prototypes are hyperspheres as shown in Fig. 1. When the Mahalanobis distance is used, the volume prototypes are hyperellipsoids.

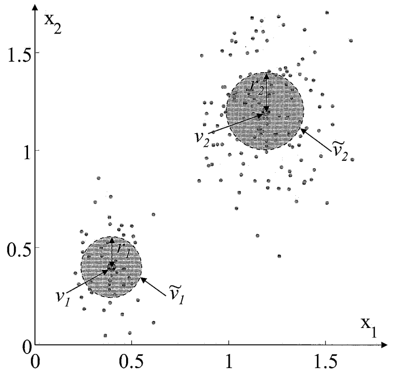


Fig. 1. Example of two E-FCM volume cluster prototypes, $\tilde{\mathbf{v}}_1$ and $\tilde{\mathbf{v}}_2$, determined from data. The cluster centers, v_1 and v_2 and the radii, r_1 and r_2, determine the position and the size, respectively, of the hyperspheres.

B. Update Equations

Clustering with volume prototypes minimizes the following objective function:

$$\tilde{J} = \sum_{i=1}^{M}\sum_{k=1}^{N} u_{ik}^{m}\left(d_{ik}^{2} - r_{i}^{2}\right) \tag{4}$$

where r_i is a constant that specifies the size of the volume prototype for cluster i. The constant $m > 1$ governs the fuzziness of the clusters found by the algorithm, while d_{ik} is the distance between the data point $\mathbf{x}_k$ and the cluster center $\mathbf{v}_i$. The goal of the clustering algorithm is to determine the cluster centers $\mathbf{v}_i$ and the membership values u_{ik} for $1 \le i \le M$, $1 \le k \le N$ by minimizing $\tilde{J}$.

The minimization of (4) subject to

$$\sum_{i=1}^{M} u_{ik} = 1 \qquad \forall k \tag{5}$$

can be achieved by alternating optimization as in regular objective function-based fuzzy clustering algorithms. The optimal update equations are obtained from the Lagrange method by setting the partial derivative of the Lagrangian with respect to $\mathbf{v}_i$ and with respect to u_{ik} equal to zero. By setting $\partial\tilde{J}/\partial\mathbf{v}_i$ equal to zero, one obtains the update equation for $\mathbf{v}_i$ as

$$\mathbf{v}_i = \frac{\sum\limits_{k=1}^{N} u_{ik}^{m}\mathbf{x}_k}{\sum\limits_{k=1}^{N} u_{ik}^{m}}. \tag{6}$$

From $\partial\tilde{J}/\partial u_{ik}$ and by eliminating the Lagrange multipliers, the update equation for u_{ik} is obtained as

$$u_{ik} = \frac{1}{\sum\limits_{j=1}^{M}\left[\frac{d_{ik}^{2}\left(1-\frac{r_i^2}{d_{ik}^2}\right)}{d_{jk}^{2}\left(1-\frac{r_j^2}{d_{jk}^2}\right)}\right]^{1/(m-1)}}. \tag{7}$$

Therefore, one could argue that the membership values in clustering with volume prototypes are obtained from the distance of the data points to the cluster centers modified by a weight factor that depends on that distance. In order to prevent membership values that are smaller than zero or larger than one, we constrain the weight factor to be nonnegative. Then, the update equation for the membership values becomes

$$u_{ik} = \frac{1}{\sum\limits_{j=1}^{M}\left[\frac{d_{ik}^{2}\sigma_{ik}}{d_{jk}^{2}\sigma_{jk}}\right]^{1/(m-1)}} \tag{8}$$

where the weighting factors σ_{ik} are given by

$$\sigma_{ik} = \max\left(0, 1 - \frac{r_i^2}{d_{ik}^2}\right). \tag{9}$$

Note that the term $d_{ik}^2\sigma_{ik}$ can also be written as a modified (squared) distance $\max(0, d_{ik}^2 - r_i^2)$.

When (4) is minimized by iterating between (6) and (8), the volume prototypes extend a distance r_i from the cluster centers and the points $\mathbf{x}_k$ within the volume prototypes are assigned a membership of one in the corresponding cluster and a membership of zero in the remaining clusters. It is possible that the data points "claim" a cluster center during the two-step optimization and lead to a suboptimal result. After all, when a number of data points are located within a cluster center, the objective function is decreased significantly due to the zero distance. This may prevent the separation of cluster centers, which normally happens in fuzzy clustering. Noting that the derivative of the Lagrangian with respect to r_i is negative for $r_i > 0$, this problem can be dealt with by setting the radii to small values initially and then gradually increasing their values to the full values specified by the user. In our method, the cluster radii are multiplied by a factor $\beta^{(l)}/M^{(l)}$, where $M^{(l)}$ is the number of clusters in the partition at iteration l of the clustering algorithm. The algorithm starts with $\beta^{(0)} = 1$. As cluster merging takes place, the size of the volume prototypes is allowed to increase by increasing the value of $\beta^{(l)}, \beta^{(l)} \le M^{(l)}$.

C. Determining the Number of Clusters

The determination of the number of "natural" groups in the data is important for the successful application of fuzzy clustering methods. We propose a similarity-based cluster merging approach for this purpose (similar approaches can be found in [13] and [14]). The method initializes the clustering algorithm with an estimated upper limit on the number of clusters. After evaluating the cluster similarity, similar clusters are merged if the similarity between clusters is higher than a threshold $\alpha \in [0, 1]$. Unlike the supervised fuzzy clustering (S-FC) approach proposed in [10], similarity-driven cluster merging does not require an additional optimization problem to be solved during clustering. Instead, a suitable similarity threshold must be selected for merging.

The goal in clustering is to obtain well-separated clusters. Inclusion measure between two fuzzy sets is an appropriate measure for assessing the similarity of fuzzy clusters. Given two fuzzy clusters $u_i(\mathbf{x}_k)$ and $u_j(\mathbf{x}_k)$, defined pointwise on $\mathbf{X}$, the fuzzy inclusion similarity measure between two fuzzy clusters is defined as [15], [16]

$$S_{ij} = \frac{\sum\limits_{k=1}^{N}\min\left(u_{ik}, u_{jk}\right)}{\min\left(\sum\limits_{k=1}^{N}u_{ik}, \sum\limits_{k=1}^{N}u_{jk}\right)}. \tag{10}$$

This measure takes into account the contribution to similarity from all data points, both from those within the volume prototypes and those outside.

The threshold $\alpha \in [0,1]$ above which merging takes place depends on the characteristics of the data set (separation between groups, cluster density, cluster size, etc.) and the clustering parameters such as the fuzziness m. In general, the merging threshold is an additional user-defined parameter for the extended clustering algorithm. The degree of similarity for two clusters also depends on the other clusters in the partition. This is due to the fact that the sum of membership for a data object is constrained to one. For the case where the selection of the threshold is problematic, we propose to use an adaptive threshold depending on the number of clusters in the partition at any time. It has been observed empirically that the adaptive threshold works best when the expected number of clusters in the data is relatively small (less than ten).

We propose to use

$$\alpha^{(l)} = \frac{1}{M^{(l)} - 1} \tag{11}$$

as the adaptive threshold. Clusters are merged when the change of maximum cluster similarity from iteration $(l-1)$ to iteration (l) is below a predefined threshold ϵ_1 and the similarity is above the threshold α. Only the most similar pair of clusters is merged and the number of clusters decreases at most one at each merger. In case of ties regarding the similarity, they are resolved arbitrarily. The algorithm terminates when the change in the elements of the partition matrix is below a defined threshold ϵ_2 (termination criterion).

III. Extended GK and FCM Algorithms

In this section, we give an algorithm for the extended fuzzy c-means (E-FCM) and the extended GK (E-GK) clustering. First, we derive an expression for the radii of the volume prototypes. Second, the algorithm with the adaptive similarity threshold (11) is given in Section III-A. The E-GK and the E-FCM algorithms differ only in step 3 of Section III-A.

Gustafson-Kessel have proposed to restrict the determinant of the norm matrix to 1, i.e., $|\mathbf{A}_i| = 1.0$. Then the norm matrix is given by

$$\mathbf{A}_i = |\mathbf{P}_i|^{1/n}\mathbf{P}^{-1}. \tag{12}$$

Using (3), the size of the cluster prototypes is calculated as

$$\mathbf{R}_i = \sqrt{\mathbf{\Lambda}_i}\mathbf{Q}_i^T|\mathbf{P}_i|^{1/n}\mathbf{Q}_i\mathbf{\Lambda}_i^{-1}\mathbf{Q}_i^T\mathbf{Q}_i\sqrt{\mathbf{\Lambda}_i} = |\mathbf{P}_i|^{1/n}\mathbf{I}. \tag{13}$$

Hence, the radius for the volume prototype is determined from the cluster volume as

$$r_i = \sqrt{|\mathbf{P}_i|^{1/n}}. \tag{14}$$

In case of the FCM algorithm, the norm matrix is the identity matrix. Applying (3) for the size of the cluster prototypes one obtains

$$\mathbf{R}_i = \sqrt{\mathbf{\Lambda}_i}\mathbf{Q}_i^T\mathbf{I}\mathbf{Q}_i\sqrt{\mathbf{\Lambda}_i} = \mathbf{\Lambda}_i. \tag{15}$$

Hence, different values for the radius are obtained depending on the direction one selects. In general, a value between the minimal and the maximal diagonal elements of $\mathbf{\Lambda}_i$ could be used as the radius. The selection of the mean radius thus corresponds to an averaging operation. The generalized averaging operator

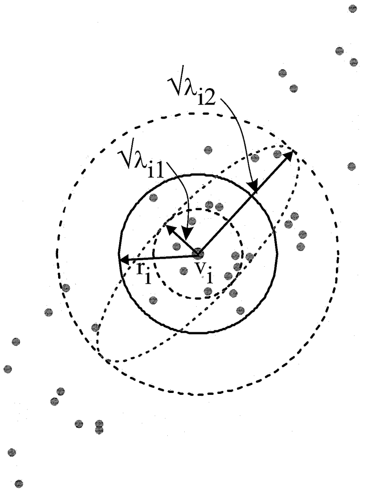


Fig. 2. The cluster volume and the E-FCM radius for a two-dimensional example.

$$D_i(s) = \frac{1}{n}\left\{\sum_{j=1}^{n}\lambda_{ij}^{s}\right\}^{1/s}, \qquad s \in \mathbb{R} \tag{16}$$

could be used for this purpose [17]. Different averaging operators are obtained by selecting different values of s in (16), which controls the bias of the aggregation to the size of λ_{ij}. For $s \rightarrow -\infty$, (16) reduces to the minimum operator and, hence, the volume prototype becomes the largest hypersphere that can be enclosed within the cluster volume (hyperellipsoid) as shown in Fig. 2. For $s \rightarrow \infty$, the maximum operator is obtained and, hence, the volume prototype becomes the smallest hypersphere that encloses the cluster volume (hyperellipsoid). It is known that the unbiased aggregation for measurements in a metric space is obtained for $s \rightarrow 0$ [18]. The averaging operator (16) then reduces to the geometric mean, so that the prototype radius is given by

$$r_i = \sqrt{\prod_{j=1}^{n}\lambda_{ij}^{1/n}} = \sqrt{|\mathbf{P}_i|^{1/n}}. \tag{17}$$

Hence, this selection for the radius leads to a spherical prototype that preserves the volume of the cluster.

Extended Clustering Algorithm: Given the data $\mathbf{X}$, choose the initial number of clusters $1 < M^{(0)} < N$, the fuzziness parameter $m > 1$ and the termination criteria $\epsilon_1, \epsilon_2 > 0$. Initialize $\mathbf{U}^{(0)}$ (e.g., random) and let $S_{i^*j^*}^{(0)} = 1$, $\beta^{(0)} = 1$.

```
Repeat for $l = 1, 2, \ldots$
1. Compute pointwise cluster prototypes (6):
   $$\mathbf{v}_i^{(l)} = \frac{\sum_{k=1}^{N}\left(u_{ik}^{(l-1)}\right)^m \mathbf{x}_k}{\sum_{k=1}^{N}\left(u_{ik}^{(l-1)}\right)^m}, \qquad 1 \le i \le M^{(l-1)}.$$
```

```
2. Compute radius of cluster prototypes
from fuzzy covariance:
    $\mathbf{P}_i = \frac{\Sigma_{k=1}^{N} \left(u_{ik}^{(l-1)}\right)^m \left(\mathbf{x}_k - \mathbf{v}_i^{(l)}\right) \left(\mathbf{x}_k - \mathbf{v}_i^{(l)}\right)^T}{\Sigma_{k=1}^{N} \left(u_{ik}^{(l-1)}\right)^m}$
    $r_i = \beta^{(l-1)} \frac{\sqrt{|\mathbf{P}_i|^{1/n}}}{M^{(l-1)}}, \qquad 1 \le i \le M^{(l-1)}.$
3. Compute the squared distances to the
volume cluster prototypes.
For extended GK clustering,
    $\tilde{d}_{ik}^2 = \max\left(0, \frac{\left(\mathbf{x}_k - \mathbf{v}_i^{(l)}\right)^T \mathbf{P}_i^{-1} \left(\mathbf{x}_k - \mathbf{v}_i^{(l)}\right)}{|\mathbf{P}_i|^{1/n}} - r_i^2\right)$
    $1 \le i \le M^{(l-1)}, \qquad 1 \le k \le N.$
For extended fuzzy c-means clustering,
    $\tilde{d}_{ik}^2 = \max\left(0, \left(\mathbf{x}_k - \mathbf{v}_i^{(l)}\right)^T \left(\mathbf{x}_k - \mathbf{v}_i^{(l)}\right) - r_i^2\right)$
    $1 \le i \le M^{(l-1)}, \qquad 1 \le k \le N.$
4. Update the partition matrix (8):
for $1 \le k \le N$, let $\phi_k = \left\{ i | \tilde{d}_{ik}^2 = 0 \right\}$
if $\phi_k = \emptyset$,
    $u_{ik}^{(l)} = \frac{1}{\sum_{j=1}^{M^{(l-1)}} \left(\frac{\tilde{d}_{ik}^2}{\tilde{d}_{jk}^2}\right)^{1/(m-1)}}, \qquad 1 \le i \le M^{(l-1)}$
otherwise
    $u_{ik}^{(l)} = \begin{cases} 0 & \texttt{if } \tilde{d}_{ik}^2 > 0 \\ \frac{1}{|\phi_k|} & \texttt{if } \tilde{d}_{ik}^2 = 0 \end{cases} \qquad 1 \le i \le M^{(l-1)}.$
5. Select the most similar cluster pair:
    $S_{ij}^{(l)} = \frac{\sum_{k=1}^{N} \min\left(u_{ik}^{(l)}, u_{jk}^{(l)}\right)}{\min\left(\sum_{k=1}^{N} u_{ik}^{(l)}, \sum_{k=1}^{N} u_{jk}^{(l)}\right)}$
    $1 \le i, j \le M^{(l-1)}$
    $(i^*, j^*) = \arg\max_{i,j<i} \left(S_{i,j}^{(l)}\right).$
6. Merge the most similar clusters:
If $\left|S_{i^*j^*}^{(l)} - S_{i^*j^*}^{(l-1)}\right| < \epsilon_1$
 let $\alpha^{(l)} = 1/(M^{(l-1)} - 1)$
 if $S_{i^*j^*}^{(l)} > \alpha^{(l)}$
      $u_{i^*k}^{(l)} := \left(u_{i^*k}^{(l)} + u_{j^*k}^{(l)}\right), \qquad 1 \le k \le N$
            remove row $j^*$ from $\mathbf{U}$
      $M^{(l)} = M^{(l-1)} - 1$
 else enlarge volume prototype
      $\beta^{(l)} = \min\left(M^{(l-1)}, \beta^{(l-1)} + 1\right).$
until $\|\mathbf{U}^{(l)} - \mathbf{U}^{(l-1)}\| < \epsilon_2$.
```

TABLE I
DATA GROUP CENTERS (x, y), VARIANCE (σ_x^2, σ_y^2) AND SAMPLE SIZE

Group	center	Variance	Number of samples
1	$(-0.5, -0.4)$	$(0.2, 0.2)$	150
2	$(0.1, 0.2)$	$(0.1, 0.1)$	30
3	$(0.5, 0.7)$	$(0.2, 0.1)$	30
4	$(0.6, -0.3)$	$(0.2, 0.25)$	50

IV. EXAMPLES

A real-world application of the E-FCM algorithm to a data mining and modeling problem in database marketing has been described in [11]. In this section, we consider the application of the extended clustering algorithms to artificially generated two-dimensional data and a multidimensional data set from the UCI Machine Learning Repository [19]. The examples illustrate the properties of the extended algorithms described in Section III. All examples have been calculated with a fuzziness parameter $m = 2$ and the adaptive threshold (11). The criterion ϵ_1 is set to 0.01 and the termination criterion ϵ_2 is set to 0.001.

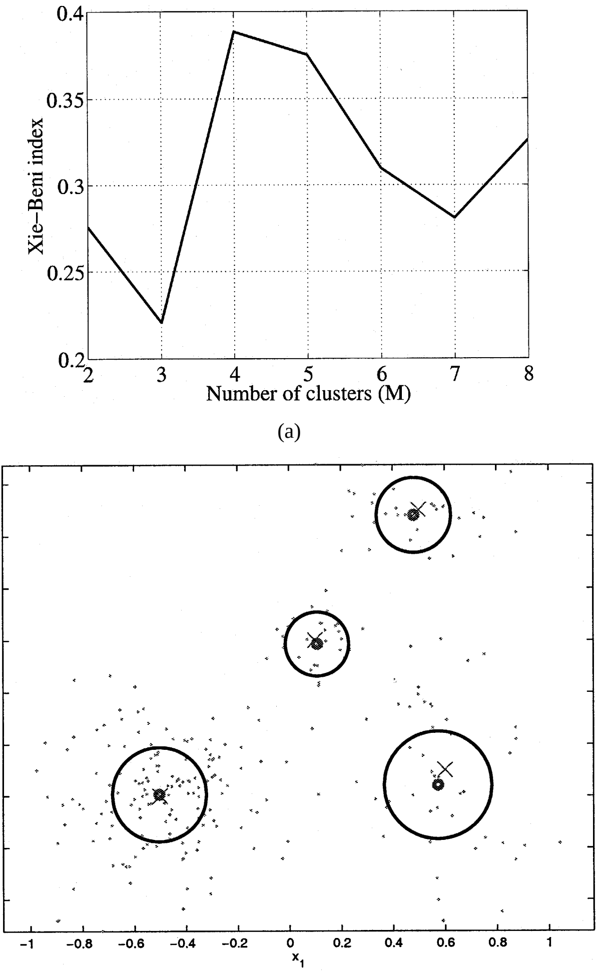
A. E-FCM Versus Cluster Validity

We want to compare the performance of an extended clustering algorithm against a cluster validity approach for discovering the underlying data structure. Four groups of data are generated randomly from normal distributions around four centers with the standard deviations given in Table I. The number of sample points in each group is also indicated. The goal is to automatically detect clusters in the data that reflect the underlying data structure. Since the clusters are roughly spherical, FCM, and E-FCM algorithms are applied.

For the cluster validity approach, the FCM algorithm is applied to the data several times with the number of clusters varying from two to eight. The resulting partitions are evaluated with the Xie–Beni cluster validity index [7], which is one of the popular cluster validity indices from the literature. The conventional approach, using the FCM algorithm and the cluster validity measure, fails to determine the correct number of structures in the data due to the uneven distribution of data [see Fig. 3(a)]. The E-FCM algorithm, however, is able to detect the four groups present in the data. The results are shown in Fig. 3(b). The E-FCM algorithm is thus more robust to uneven data distribution than the original FCM algorithm. However, the E-FCM algorithm can also lead to wrong results if the differences in cluster density become too large, as shown in Fig. 4, where cluster 1 has now 300 points.

B. Influence of Initialization

To study the influence of initialization on the extended clustering algorithms, the data in Section IV-A is clustered 1000 times both with the FCM and the E-FCM algorithms. The partitions have been initialized randomly each time. The FCM algorithm is set to partition the data into four clusters, while the E-FCM algorithm is started with ten clusters initially. After each run, the cluster centers are recorded. Table II shows the mean cluster centers and the standard deviation of the cluster



(a)

(b)

Fig. 3. (a) Combination of FCM and cluster validity fails in determining the four groups in the data set. (b) The E-FCM algorithm automatically detects the correct number of data structures in the data set. The data (·), group centers (x) and E-FCM cluster centers (o) are shown.

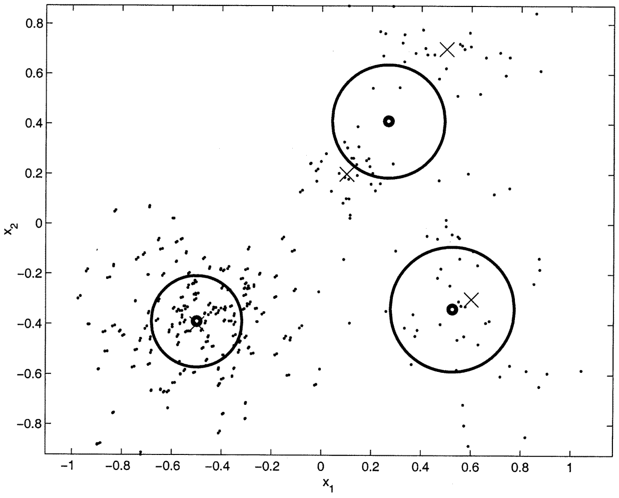


Fig. 4. An example where E-FCM algorithm converges to a nonintuitive solution because of very large differences in the cluster densities.

TABLE II
MEAN AND STANDARD DEVIATION OF CLUSTER CENTERS FOUND BY THE FCM AND E-FCM ALGORITHMS AFTER 1000 EXPERIMENTS WITH RANDOM INITIALIZATION

		FCM center		E-FCM center	
Group	True center	Mean	Std. dev.	Mean	Std. dev.
1	(-0.5,-0.4)	(-0.61,-0.44)	(0.034,0.013)	(-0.50,-0.39)	$< 10^{-5}$
2	(0.1,0.2)	(-0.29,-0.26)	(0.134,0.142)	(0.11,0.18)	$< 10^{-5}$
3	(0.5,0.7)	(0.39,0.56)	(0.030,0.035)	(0.48,0.68)	$< 10^{-5}$
4	(0.6,-0.3)	(0.55,-0.33)	(0.004,0.024)	(0.57,-0.36)	$< 10^{-5}$

center coordinates after 1000 experiments. It is observed that the cluster centers found by the E-FCM algorithm are closer to the true centers than the ones found by the FCM algorithm. Moreover, the standard deviation of the centers is much lower for the E-FCM algorithm. The FCM algorithm especially has difficulty with the small data group 2, which seems to be missed if the initialization is not good. Therefore, the mean cluster center is far away from any of the true cluster centers and the standard deviation of the center coordinates is very large. The E-FCM algorithm has proven to be much more robust to the partition initialization. In fact, the similarity threshold α has a larger impact on the algorithm than initialization. This is to be expected since merging too many or too few clusters would change the remaining center coordinates significantly.

C. Computational Load

When using volume prototypes, the computational load of the E-FCM algorithm without cluster merging is larger than the computational load of the FCM algorithm. This is partly caused by the slower convergence due to the cluster radii that are enlarged gradually during the optimization. Similarly, the computational load of the E-GK algorithm without cluster merging is larger than the computational load of the GK algorithm. In general, the computational complexity of the GK algorithms is larger than the computational complexity of the FCM algorithms due to the additional calculations of cluster covariance and the inversion of the covariance matrix for use in the distance metric. When (17) is used to compute the cluster radii in E-FCM, however, the cluster covariance matrix must also be computed, which increases the computational load of the E-FCM algorithm compared to the original FCM algorithm even further. Naturally, the computational cost of the extended algorithms increases further when cluster merging is used. In this case, the clustering is made multiple times for different number of clusters and hence the total clustering time depends on the initial number of clusters $M^{(0)}$.

In order to compare the computational load of various algorithms, we have run different algorithms 100 times with the data set from Section IV-B. Each time, the cluster algorithms are initialized randomly. When cluster merging is applied, the algorithms are started with ten initial clusters. Table III summarizes the results obtained on a 650-MHz Pentium III machine with 256-MB memory running Matlab. As Table III shows, the extended algorithms are three to four times slower than the corresponding original clustering algorithms.

D. Line Detection

The E-GK algorithm is capable of determining differently shaped clusters in the same data set. Fig. 5 shows the application

TABLE III
COMPUTATIONAL LOAD AVERAGED OVER 100 DIFFERENT INITIALIZATIONS

algorithm	FCM	GK
original	10.4s	14.3s
extended without cluster merging	30.0s	54.3s
extended with cluster merging	87.1s	146.3s

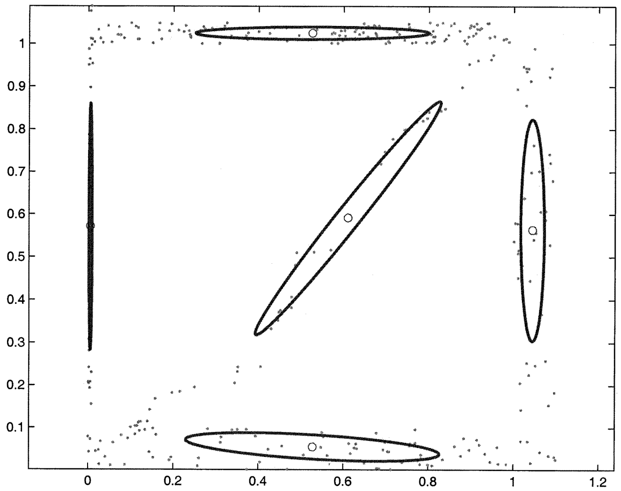


Fig. 5. The E-GK algorithm correctly identifies the five noisy lines in the data set. The algorithm is initialized with ten clusters.

TABLE IV
CONFUSION MATRIX FOR E-FCM CLASSIFYING THE WINE DATA

	Classified by E-FCM			
actual	class 1	class 2	class 3	total
class 1	59	0	0	59
class 2	3	65	3	71
class 3	0	0	48	48
total	62	65	51	178

of the E-GK algorithm to a data set with five noisy linear data groups. The algorithm is initialized with ten clusters. It automatically detects the five groups in the data. Note how the volume prototypes are adjusted to the various thickness of the lines.

E. Wine Data

To illustrate the performance of the extended fuzzy c-means algorithm in a higher dimensional problem, we have applied it on the wine database from the UCI Machine Learning Repository [19]. In this data set, three classes of wine are described by 13 different features regarding the chemical composition of the wine. All features are continuous valued. The E-FCM algorithm is applied starting with ten clusters. The merge threshold α is 0.70, which leads to three clusters that we expect to find in this data set. Unsupervised classification based on the clustering results of E-FCM leads to an overall accuracy of 97%. The confusion matrix corresponding to this classification is given in Table IV.

V. CONCLUSION

Two extensions have been proposed to the objective function-based fuzzy clustering algorithms in order to deal with some critical issues in fuzzy clustering. The extensions consist of the use of volume cluster prototypes and similarity-driven merging of clusters. The volume prototypes imply that the data points close to a cluster center are assumed to belong fully to that cluster. Similarity-driven merging helps determine a suitable number of clusters starting from an overestimated number of clusters. By initializing the clustering algorithm with an overestimated number of clusters, the possibility increases for the algorithm to detect all the important regions of the data. This decreases the dependency of the clustering result on the (random) initialization.

Extended version of the fuzzy c-means and the GK clustering algorithms is given. It is shown that clustering with volume prototypes can be formulated as the minimization of an objective function. The cluster merging step is motivated heuristically. Merging results depend on the selection of a similarity threshold, which is a disadvantage of the proposed method. However, an adaptive similarity threshold is proposed that alleviates this problem partially for some data sets. In several examples, we have also shown that the proposed algorithms are capable of determining a suitable partition of the data.

ACKNOWLEDGMENT

The authors would like to thank the anonymous referees for their comments and valuable suggestions.

REFERENCES

[1] F. Höppner, F. Klawonn, R. Kruse, and T. Runkler, *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition*. New York: Wiley, 1999.

[2] J. C. Dunn, "A fuzzy relative of the isodata process and its use in detecting compact, well-seperated clusters," *J. Cybern.*, vol. 3, no. 3, pp. 32–57, 1973.

[3] D. E. Gustafson and W. C. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," in *Proc. IEEE Conf. Decision Control*, San Diego, CA, 1979, pp. 761–766.

[4] J. C. Bezdek, *Pattern Recognition With Fuzzy Objective Function*. New York: Plenum, 1981.

[5] R. Krishnapuram and J. Kim, "Clustering algorithms based on volume criteria," *IEEE Trans. Fuzzy Syst.*, vol. 8, pp. 228–236, Apr. 2000.

[6] I. Gath and A. B. Geva, "Unsupervised optimal fuzzy clustering," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 773–781, July 1989.

[7] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 841–847, Aug. 1991.

[8] R. Krishnapuram and C.-P. Freg, "Fitting an unknown number of lines and planes to image data through compatible cluster merging," *Patt. Recog.*, vol. 25, no. 4, pp. 385–400, 1992.

[9] U. Kaymak and R. Babuška, "Compatible cluster merging for fuzzy modeling," in *Proc. Fourth IEEE Int. Conf. Fuzzy Systems*, vol. 2, Yokohama, Japan, Mar. 1995, pp. 897–904.

[10] M. Setnes, "Supervised fuzzy clustering for rule extraction," in *Proc. FUZZ-IEEE'99*, Seoul, Korea, Aug. 1999, pp. 1270–1274.

[11] M. Setnes and U. Kaymak, "Fuzzy modeling of client preference from large data sets: An application to target selection in direct marketing," *IEEE Trans. Fuzzy Syst.*, vol. 9, pp. 153–163, Feb. 2001.

[12] A. Keller and F. Klawonn, "Clustering with volume adaptation for rule learning," in Proc. Seventh Euro. Congr. Intelligent Techniques Soft Computing (EUFIT'99), Aachen, Germany, Sept. 1999.

[13] H. Frigui and R. Krishnapuram, "A robust algorithm for automatic extraction of an unknown number of clusters from noisy data," *Patt. Recog. Lett.*, vol. 17, pp. 1223–1232, 1996.

[14] E. Backer and A. K. Jain, "A clustering performance measure based on fuzzy set decomposition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-3, pp. 66–75, Jan. 1981.

[15] D. Dubois and H. Prade, "A unifying view of comparison indices in a fuzzy set-theoretic framework," in *Fuzzy Sets and Possibility Theory: Recent Developments*, R. R. Yager, Ed. New York: Pergamon, 1982, pp. 3–13.

[16] B. Kosko, *Neural Networks and Fuzzy Systems*. Upper Saddle River, NJ: Prentice-Hall, 1992.

[17] G. H. Hardy, J. E. Littlewood, and G. Polya, *Inequalities*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1973.

[18] U. Kaymak and H. R. van Nauta Lemke, "A sensitivity analysis approach to introducing weight factors into decision functions in fuzzy multicriteria decision making," *Fuzzy Sets Syst.*, vol. 97, no. 2, pp. 169–182, July 1998.

[19] C. L. Blake and C. J. Merz, *UCI Repository of Machine Learning Databases*, 1998.

Uzay Kaymak received the M.Sc. degree in electrical engineering, the Degree of Chartered Designer in Information Technology, and the Ph.D. degree, all from Delft University of Technology, Delft, The Netherlands, in 1992, 1995, and 1998, respectively.

He worked between 1997–2000 as a Reservoir Engineer at Shell International Exploration and Production, The Netherlands. Currently, he is an Assistant Professor at Erasmus University Rotterdam, Rotterdam, The Netherlands. His research interests include fuzzy decision making, data mining for marketing and finance, and intelligent agents for financial modeling.

Dr. Kaymak is a Member of the Dutch School for Information and Knowledge Systems (SIKS) and of the Erasmus Research Institute for Management (ERIM).

Magne Setnes was born in 1970, in Bergen, Norway. He received the B.Sc. degree in robotics from the Kongsberg College of Engineering, Norway, the M.Sc. degree in electrical engineering from the Delft University of Technology, Delft, The Netherlands, in 1992 and 1995, respectively. He also received the Degree of Chartered Designer in Information Technology and the Ph.D. degree from the Control Laboratory, both from Delft University, in 1997 and 2001, respectively.

Currently, he is with Heineken Technical Services, Department of Research & Development, The Netherlands. His interests include fuzzy systems and computational intelligence techniques for modeling, control, and decision making.