Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

PERGAMON

Pattern Recognition 37 (2004) 2027–2037

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

FCM-Based Model Selection Algorithms for Determining the Number of Clusters

Haojun Sun[a], Shengrui Wang[a,*], Qingshan Jiang[b]

[a] *Department of Computer Science, Faculty of Sciences, University of Sherbrooke, Sherbrooke, QC, Canada, J1K 2R1*
[b] *Department of Computer Science, Xiamen University, Fujian 361005, China*

Received 16 December 2002; received in revised form 29 March 2004; accepted 29 March 2004

Abstract

Clustering is an important research topic that has practical applications in many fields. It has been demonstrated that fuzzy clustering, using algorithms such as the fuzzy C-means (FCM), has clear advantages over crisp and probabilistic clustering methods. Like most clustering algorithms, however, FCM and its derivatives need the number of clusters in the given data set as one of their initializing parameters. The main goal of this paper is to develop an effective fuzzy algorithm for automatically determining the number of clusters. After a brief review of the relevant literature, we present a new algorithm for determining the number of clusters in a given data set and a new validity index for measuring the "goodness" of clustering. Experimental results and comparisons are given to illustrate the performance of the new algorithm.

Keywords: Clustering; Fuzzy C-means; Validity index; Overlapping clusters

1. Introduction

Clustering is a process for grouping a set of objects into classes or clusters so that the objects within a cluster have high similarity, but are very dissimilar to objects in other clusters [1]. Various types of clustering methods have been proposed in the literature [2–7]. All of these methods share a common feature: they are unsupervised. Because of this, the clustering results need to be validated. The cluster validation problem involves measuring how well the clustering results reflect the structure of the data set, which is an important issue in cluster analysis. The most important indicator of the structure is the number of clusters. Since most basic clustering algorithms assume that the number of clusters in a data set is a user-defined parameter (one that is difficult to set in practical applications), the common approach is an iterative trial-and-error process. The trial-and-error process performs the model selection according to the terms used by Jain [5]. In fact, the number of clusters is a parameter related to the complexity of the cluster structure. In this paper, we are interested in the problem of cluster validation in the context of partitioning-based clustering algorithms. In other words, the clustering algorithm is run with different initial values for the number of clusters and the results are compared in order to determine the most appropriate number of clusters. For this purpose, validity indices have been proposed in the literature [8–12].

In the work reported here, we are particularly interested in the fuzzy C-means (FCM) algorithm. Because of its concept of fuzzy membership, FCM is able to deal more effectively with outliers and to perform membership grading, which is very important in practice. FCM is one of the most widely used clustering algorithms. Several validity indices have been proposed in the literature for use with the FCM clustering algorithm. Early indices such as the partition coefficient and classification entropy make use only of membership values and have the advantage of being easy to compute. Now, it is widely accepted that a better definition of a validity index must consider both the compactness within each cluster and the separation between clusters.

* Corresponding author.
E-mail addresses: sun@dmi.usherb.ca (H. Sun), wang@dmi.usherb.ca (S. Wang), qjiang@xmu.edu.cn (Q. Jiang).

0031-3203/$30.00 © 2004 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2004.03.012

Most existing validity indices are efficient in detecting the number of clusters when the data in different clusters do not overlap. However, in Section 5.2, we will see that for overlapping data, their behavior could be unpredictable.

In this paper, we report two contributions to cluster analysis. First, we propose a new algorithm for clustering while automatically determining the number of clusters. The new algorithm improves the conventional model selection process by reducing the randomness in the initialization of cluster centers at the beginning of each clustering phase. For this purpose, splitting strategies have been designed and combined with the basic clustering algorithm so that for each candidate number of clusters, the clustering process can be carried out starting with the previously obtained clusters. Second, we propose a new validity index that is efficient even when clusters overlap each other. The new validity index, inspired by Rezaee's validity, is based on a linear combination of compactness and separation. We report the test results yielded by the index in a model selection process using a data set from the public domain and several generated data sets. These results provide an evaluation of the new index under the condition of overlapping clusters, an empirical comparison with other indices, and an evaluation of new algorithm in terms of numerical stability and running time cost.

This paper is organized as follows. In Section 2, we introduce the basic FCM algorithm and the FCM-based model selection algorithm. In Section 3, we present the new algorithm, which is based on a splitting strategy. In Section 4, we describe our new validity index in detail. In Section 5, experimental results are presented to show the advantages of the new algorithm and the new validity index. In Section 6 we give some concluding remarks and discuss current and future extensions of this work.

2. Basic algorithm

In this section, we briefly introduce the basic FCM algorithm and the general model selection algorithm for determining the number of clusters in a data set.

2.1. FCM algorithm

The FCM algorithm dates back to 1973. FCM-based algorithms are the most widely used fuzzy clustering algorithms in practice. The basic FCM algorithm can be formulated as follows:

$$Minimize\, J_m(U,V)=\sum_{k=1}^{n}\sum_{i=1}^{c}u_{ki}^{m}\|x_k-v_i\|^2, \tag{1}$$

where n is the total number of data vectors in a given data set and c is the number of clusters; $X=\{x_1,x_2,\ldots,x_n\}\subset R^s$ and $V=\{v_1,v_2,\ldots,v_c\}\subset R^s$ are the feature data and cluster centers; and $U=(u_{ki})_{n\times c}$ is a fuzzy partition matrix composed of the membership of each feature vector x_k in each cluster i. u_{ki} should satisfy $\sum_{i=1}^{c}u_{ki}=1$ for $k=1,2,\ldots,n$ and $u_{ki}\geqslant 0$ for all $i=1,2,\ldots,c$ and $k=1,2,\ldots,n$. The exponent $m>1$ in $J_m(U,V)$ (Eq. (1)) is a parameter, usually called a fuzzifier. To minimize $J_m(U,V)$, the cluster centers (prototypes) v_i and the membership matrix U need to be computed according to the following iterative formula:

$$u_{ki}=\left\{\begin{array}{ll}\left(\displaystyle\sum_{j=1}^{c}\left(\frac{\|x_k-v_i\|}{\|x_k-v_j\|}\right)^{\frac{2}{m-1}}\right)^{-1} & \text{if } \|x_k-v_j\|>0,\\ & \forall j,\\ 1 & \text{if } \|x_k-v_i\|=0\\ 0 & \text{if } \exists j\neq i\ \|x_k-v_j\|\\ & =0,\end{array}\right\}, \tag{2}$$

For $K=1,\ldots,n$ and $i=1,\ldots,c$.

$$v_i=\frac{\sum_{k=1}^{n}u_{ki}^{m}x_k}{\sum_{k=1}^{n}u_{ki}^{m}},\quad i=1,2,\ldots,c. \tag{3}$$

The basic FCM algorithm is as follows.

Algo1: Basic FCM algorithm

(1) Input the number of clusters c, the fuzzifier m and the distance function $\|\cdot\|$.
(2) Initialize the cluster centers $v_i^0(i=1,2,\ldots,c)$.
(3) Calculate $u_{ki}(k=1,2,\ldots,n;i=1,2,\ldots,c)$ using Eq. (2).
(4) Calculate $v_i^1(i=1,2,\ldots,c)$ using Eq. (3).
(5) If $\max_{1\leqslant i\leqslant c}(\|v_i^0-v_i^1\|/\|v_i^1\|)\leqslant\varepsilon$ then go to *Step* 6; else let $v_i^0=v_i^1(i=1,2,\ldots,c)$ and go to *Step* 3.
(6) Output the clustering results: cluster centers $v_i^1(i=1,2,\ldots,c)$, membership matrix U and, in some applications, the elements of each cluster i, i.e., all the x_k such that $u_{ki}>u_{kj}$ for all $j\neq i$.
(7) Stop.

2.2. Determination of the number of clusters

The following model selection algorithm applies the basic FCM clustering algorithm to the data set for $c=C_{min},\ldots,C_{max}$ and chooses the best value of c based on a (cluster) validity criterion. Here, C_{min} and C_{max} are predefined values that represent, respectively, the minimal and maximal numbers of clusters between which an optimal number is sought.

Algo2: FCM-based model selection algorithm

(1) Choose C_{min} and C_{max}.
(2) For $c=C_{min}$ to C_{max}:
  (2.1) Initialize cluster centers (V).
  (2.2) Apply the basic FCM algorithm to update the membership matrix (U) and the cluster centers (V).

(2.3) Test for convergence; if no, go to 2.2.
(2.4) Compute a validity value $V_d(c)$.
(3) Compute c_f such that the cluster validity index $V_d(c_f)$ is optimal.

Several techniques exist for initializing cluster centers (Step 2.1). Random initialization is often used because of its simplicity. Other initialization methods could be used in many cases. Recently, an empirical comparison of four initialization methods for the K-Means algorithm was reported in Ref. [13]. According to this study, random initialization is one of the best methods as it makes the K-Means algorithm more effective and less dependent on initial clustering and order of instances. Although it is not clear if these results can be generalized to the case of FCM, it is still reasonable to assume that random initialization is a good choice for Algo2.

3. A new FCM-based clustering algorithm

In Algo2, we use random initialization at the beginning of each clustering phase. By doing so, we try to ensure that the selection process is carried out under relatively general conditions and the results are as replicable as possible. However, it is easy to imagine that re-initialization at each phase could be a source of computational inefficiency. Use of the clustering results obtained in previous phases may lead to a better initialization. In this section, we propose strategies that yield a new FCM-based clustering algorithm. First we explain the major steps of the algorithm in detail. Experimental results and discussions follow in subsequent sections.

The FCM-based splitting algorithm (FBSA) described below is called a splitting algorithm because it operates by splitting the "worst" cluster at each stage in testing the number of clusters c from C_{min} to C_{max}. The major differences between this algorithm and Algo2 in the previous section lie in the initialization of cluster centers, the validity index used and the process for splitting "bad" clusters. The general strategy adopted for the new algorithm is as follows: at each step of the new algorithm (FBSA), we identify the "worst" cluster and split it into two clusters while keeping the other $c-1$ clusters.

FBSA: FCM-Based Splitting Algorithm

(1) Choose C_{min} and C_{max}.
(2) Initialize C_{min} cluster centers (V).
(3) For $c = C_{min}$ to C_{max}:
  (3.1) Apply the basic FCM algorithm to update the membership matrix (U) and the cluster centers (V).
  (3.2) Test for convergence; if no, go to 3.1.
  (3.3) Compute a validity value $V_d(c)$.
  (3.4) Compute a score $S(i)$ for each cluster; split the worst cluster.
(4) Compute c_f such that the cluster validity index $V_d(c_f)$ is optimal.

The general idea in the splitting algorithm FBSA is to identify the "worst" cluster and split it, thus increasing the value of c by one. Our major contribution lies in the definition of the criterion for identifying the "worst" cluster. In this paper, we propose a "score" function $S(i)$ associated with each cluster i, as follows:

$$S(i) = \frac{\sum_{k=1}^{n} u_{ki}}{number\_of\_data\_vectors\_in\_cluster\_i}.$$

In general, when $S(i)$ is small, cluster i tends to contain a large number of data vectors with low membership values. The lower the membership value, the farther the object is from its cluster center. Therefore, a small $S(i)$ means that cluster i is large in volume and sparse in distribution. This is the reason we choose the cluster corresponding to the minimum of $S(i)$ as the candidate to split when the value of c is increased. On the other hand, a larger $S(i)$ tends to mean that cluster i has a smaller number of elements and exerts a strong "attraction" on them.

In order to split the cluster at Step 3.4 of FBSA, we have adapted the "Greedy" technique [14]. The "Greedy" technique aims to initialize the cluster centers as far apart from each other as possible. In an iterative manner, the "Greedy" technique selects as a new cluster center the data vector which has the largest total distance from the existing cluster centers. Adaptation of the technique for cluster splitting yields the following algorithm:

(1) Identify the cluster to be split (first part of 3.4). Supposing that the cluster number is i_0, its center and the set of all the data in the cluster are denoted by V_{i_0} and E.
(2) Search E for the data vector not labeled "tested" which has the maximal total distance from all of the remaining $c-1$ cluster centers. This data vector is denoted by V_{i_1}.
(3) Partition E into E_0 and E_1 based on the distance of each data vector from V_{i_0} and V_{i_1}. If $|E_1|/|E| > 10\%$, then V_{i_1} is taken as the c^{th} cluster center; else label V_{i_1} "tested" and go to Step 2.
(4) Search E for the data vector not labelled "tested" which has the maximal total distance from all of the c cluster centers. This data vector is denoted by V_{i_2}.
(5) Partition E into E_1 and E_2 based on the distance of the data vector from V_{i_1} and V_{i_2}. If $|E_2|/|E| > 10\%$, then V_{i_2} is taken as the $(c+1)$th cluster center, else label V_{i_2} "tested" and go to Step 4.

This algorithm ensures that the two new centers V_{i_1} and V_{i_2} are as far apart as possible from each other and from the $c-1$ centers (of the unsplit clusters). In addition, a significant number of data vectors (10% of E) are required to be in the

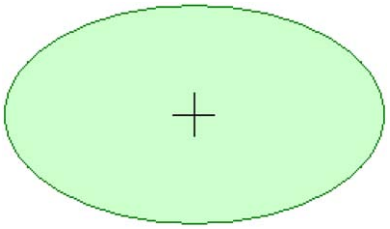

Fig. 1. Cluster before the split. The center of the cluster is marked by "+".

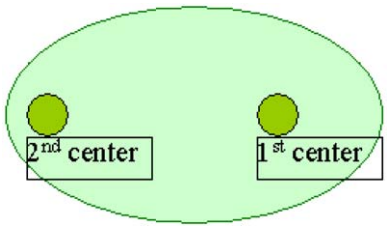

Fig. 2. Cluster after the split. The small circles here mark the initial centers of new clusters.

neighborhood of each center so as to minimize the possibility of picking up an outlier. Figs. 1 and 2 illustrate a typical result of the splitting algorithm.

4. A new validity index

The index $V_d(c_f)$ in Algo2 and FBSA measures the *goodness* of the results of a clustering algorithm. A partition is considered good if it optimizes two conflicting criteria. One of these is related to within-class scattering, which needs to be minimized; the other to between-class scattering, which needs to be maximized. The major validity indices for fuzzy clustering found in the literature are reviewed in the following section. Then, a new index is introduced. The performance of the various indices is compared in Section 5.

4.1. Validity indices for fuzzy clustering

There are a number of cluster validity indices available. Some of them use only the membership values of a fuzzy partition of the data (membership matrix), others use the original data and the computed cluster centers as well as the membership matrix. Here are some of the indices most frequently referred to in the literature.

V_{PC} (Partition coefficient) and V_{PE} (Partition entropy) [15] are two simple indices that are computed using only the elements of the membership matrix. Both indices are easy to compute. They are useful when the data contains only a small number of well-separated clusters. However, there is a lack of direct connection to the geometrical properties of the data. Xie and Beni (1991) defined a well-known validity index, V_{Xie}, which measures overall average compactness against separation of the c-partition [8]. Fakuyama and Sugeno (1989) proposed another validity index, V_{FS}, which measures the discrepancy between compactness and separation of clusters [9].

Recently, several other validity indices have been proposed, using different definitions of compactness and separation. Rhee and Ho [11] proposed an index V_{RH}, which yields good results in terms of the accuracy of the final number of clusters. However, the computational complexity for calculating the value of this index is $O(n^2c)$, making it difficult to use in practice. Zahid et al. [16] proposed an index V_{ZLE}, which considers the geometrical properties, the degree of fuzzy membership and the structure of the data. Rezaee et al. [10] proposed a validity index, V_{RLR}, which depends on a linear combination of the average scattering (compactness) of clusters and distance (separation) between clusters. Our new index was inspired by the Rezaee–Letlieveldt–Reiber index.

4.2. A new validity index

In our experiments, we found that the currently used validity indices behave poorly when clusters overlap each other. This motivated our search for an efficient validity index. The validity index we propose $V_{WSJ}(U,V,c)$ has the following form:

$$V_{WSJ}(U,V,c) = Scat(c) + \frac{Sep(c)}{Sep(C_{max})}. \tag{4}$$

Here $Scat(c) = \frac{\frac{1}{c}\sum_{i=1}^{c}\|\sigma(v_i)\|}{\|\sigma(X)\|}$, which is defined in the same way as in the Rezaee–Letlieveldt–Reiber index, where $\sigma(X) = \{\sigma(X)^1, \sigma(X)^2, \ldots, \sigma(X)^s\}^{\mathrm{T}}$, $\bar{x} = \frac{1}{n}\sum_{k=1}^{n} x_k$, $\sigma(X)^p = \frac{1}{n}\sum_{k=1}^{n}(x_k^p - \bar{x}^p)^2$, $\sigma(v_i) = \{\sigma(v_i)^1, \sigma(v)_i)^2, \ldots, \sigma(v_i)^s\}^{\mathrm{T}}$, and $\sigma(v_i)^p = \frac{1}{n}\sum_{k=1}^{n} u_{ki}(x_k^p - v_i^p)^2$, for $p = 1, 2, \ldots, s$. It represents the compactness of the obtained clusters. The value of $Scat(c)$ generally decreases when c increases because the clusters become more compact. The range of $Scat(c)$ is between 0 and 1. The term representing the separation between clusters is defined as

$$Sep(c) = \frac{D_{max}^2}{D_{min}^2}\sum_{i=1}^{c}\left(\sum_{j=1}^{c}\|v_i - v_j\|^2\right)^{-1},$$

where $D_{min} = min_{i\neq j}\|v_i - v_j\|$ and $D_{max} = max_{i,j}\|v_i - v_j\|$. Intuitively speaking, the new validity index $V_{WSJ}(U,V,c)$ improves that of Rezae–Letlieveldt–Reiber in that it provides a better balance between the two conflicting factors. Various experimental results demonstrating its efficiency will be given in the next section.

To gain some insight into this definition of separation, $Sep(c)$ can be written approximately as $Sep(c) \doteq \frac{c}{c-1}\frac{D_{max}^2}{D_{min}^2}E[\frac{1}{d_c^2}]$, where d_c is the average distance from a

Table 1
Means of DataSet2

	Cluster1	Cluster2	Cluster3	Cluster4	Cluster5
x	1.0	1.0	−0.5	−0.5	−0.5
y	0.3	−0.3	0	−0.3	0.3
z	0.0	0.0	0.5	0.0	0.0

cluster center to all the other cluster centers. Both $\frac{D_{max}^2}{D_{min}^2}$ and $E[\frac{1}{d_c^2}]$ in $Sep(c)$ are influenced by the geometry of the cluster centers. Both factors tend to be small when the cluster centers are well distributed. For example, when the cluster centers form a tetrahedron, $\frac{D_{max}^2}{D_{min}^2}$ reaches its minimum, which is 1; while $E[\frac{1}{d_c^2}]$ also reaches its minimum, $\frac{1}{D_{max}^2}$, whose value depends only on the absolute distance (which is a scale factor) between any two centers. When the distribution of cluster centers is irregular, both $\frac{D_{max}^2}{D_{min}^2}$ and $E[\frac{1}{d_c^2}]$ become larger. However, their values tend to evolve in different ways when the number of clusters c in the clustering algorithm increases. In fact, $\frac{D_{max}^2}{D_{min}^2}$ will likely increase as more (calculated) cluster centers tend to result in increased D_{max} and decreased D_{min} at the same time. On the other hand, $E[\frac{1}{d_c^2}]$ will likely become more stable as the estimate of the average distance d_c becomes more accurate. This is also why $\frac{D_{max}^2}{D_{min}^2}$ is important, since it is the main factor that penalizes model structures with too many clusters. A cluster number which minimizes $V_{WSJ}(U, V, c)$ is considered to be the optimal value for the number of clusters present in the data.

5. Experimental results

In this section, we present the performance of the new algorithm and the new validity index. We report the experimental results for four data sets, the first one from the public domain, the next two generated using mixtures of Gaussian distributions and the fourth one from a real survey data set. For each of the first three data sets, we evaluate the algorithm and index using three criterions: accuracy of clustering results, stability across different runs and time cost. The fourth data set is used only for evaluating the time efficiency of the proposed algorithm.

In all experiments, the fuzzifier m in the algorithm was set to 2, the test for convergence in the basic FCM algorithm (Algo1) was performed using $\varepsilon = 0.001$, and the distance function $\|\cdot\|$ was defined as Euclidean distance. Choosing the best range of the number of clusters is a difficult problem. Here, for the first three data sets, we adopted Bezdek's suggestion: $C_{min} = 2$ and $C_{max} = \sqrt{n}$ [15]. For determination of the number of clusters, the validity indices

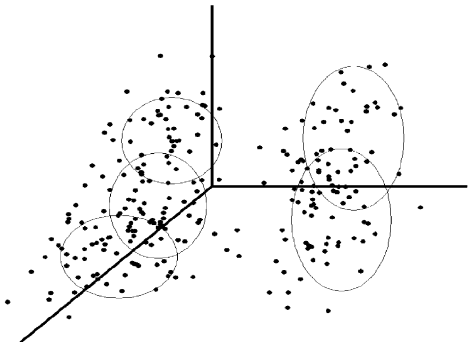

Fig. 3. DataSet2 is a 3D data set and has five clusters.

V_{PC}, V_{PE}, V_{Xie}, V_{FS}, V_{RH}, V_{ZLE} and V_{RLR} were compared with V_{WSJ}. The initialization of cluster centers in FBSA (Step 2) and Algo2 (Step 2.1) used the random procedure.

5.1. Data Sets

The first data set, DataSet1, is IRIS data set [17], widely used for testing clustering algorithms. This is a biometric data set consisting of 150 measurements belonging to three flower varieties. The data are represented as vectors in a 4-dimensional measurement space, in which four variables are length and width of both petal and sepal. The set consists of three classes, each of which contains 50 observations. In fact, of the three classes, two are overlapped. Halgamuge and Glesner [18] have shown that a very good classification can be obtained using only two features. In Ref. [10], Rezaee et al. indicate that for their index V_{RLR}, only one feature (petal length) is used to obtain the best number of classes, which is 3.

DataSet2 was generated using a mixture of Gaussian distributions. This data set is three-dimensional and contains five Gaussian components (clusters). There are 50 data vectors in each of the five clusters. For each component, the three variables are independent of each other and their variance is 0.2. The means of the five clusters (components) are given in Table 1. Fig. 3 shows the 3D picture. In the data set, Clusters 1 and 2 strongly overlap each other and Clusters 3, 4 and 5 strongly overlap each other.

DataSet3 was also generated using a mixture of Gaussian distributions too. This example contains 500

Table 2
Means and variances of DataSet3

		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
x	Mean	0.0	4.5	4.5	1.5	−2.0	−5.5	−3.5	−2.5	2.0	7.0
	Variance	1.0	1.5	2.0	1.0	1.0	1.0	0.5	0.5	0.5	0.5
y	Mean	0.0	3.0	0.0	3.0	−3.0	−1.0	2.0	4.5	−3.5	−3.5
	Variance	0.5	0.5	0.5	0.5	1.5	0.5	1.0	1.0	1.0	0.5

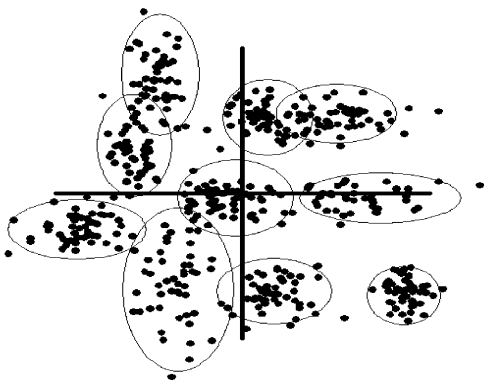

Fig. 4. DataSet3 is a 2D data set and has 10 clusters.

two-dimensional data vectors. It consists of 10 Gaussian components (clusters). For each component, the two variables are independent. There are 50 data vectors in each of the 10 clusters. The means and variances of the 10 clusters (components) are given in Table 2. This data set has been generated so that Clusters 1, 2, 3, and 4 overlap. Fig. 4 illustrates the data set.

5.2. Comparison of accuracy of clustering results

The main objective of this subsection is to compare the performance of different validity indices in determining the true number of clusters. There are two parts in this comparison: the optimal number of clusters(C_{Opt}) and the errors between cluster centers and component means. Normally, each run of the algorithm Algo2 or the algorithm FBSA involves only one validity index. However, random initialization in these algorithms may have some effect on their (average) performance given that the number of runs of each algorithm is always limited (to 20 in our experiments). In order to eliminate the disparities in performance induced by random initialization, we simply need to compute all the validity indices on the same set of the clusters, yielded by Algo2 or FBSA for $c = C_{min}$ to C_{max}, and record and compare the optimal cluster number corresponding to each validity index. For this reason, all the experiments in this section were performed with all the tested validity indices implemented within Algo2 and FBSA. With this setting, if two validity indices yield the same optimal number of clusters in a run, they yield exactly the same clusters too. This point is particularly important for understanding the results discussed in Section 5.2.2, where two validity indices can yield exactly the same average position error for cluster centers even with a significant number of random initializations in the clustering algorithms.

5.2.1. Accuracy of the optimal number of clusters(C_{Opt})

Finding the true number of clusters is a fundamental goal of the clustering algorithm. The validity index often plays a key role in model selection approaches. Here we have tested the existing well-known validity indices V_{PC}, V_{PE}, V_{Xie}, V_{FS}, V_{RH}, V_{ZLE}, V_{RLR} and our new validity index V_{WSJ} in Algo2 and FBSA. Each algorithm was run 20 times with different initial centers in order to evaluate the stability of the algorithm and the validity index used.

Tables 3–5 give the results for the optimal number of clusters (20 runs) when all validity indices are applied to Algo2 for the three data sets. In these tables, $C_{Opt}(m)$ means that the optimal value C_{Opt} was obtained m times in 20 runs. For the IRIS data set, V_{WSJ} yields an optimal number of clusters of 3 (the best number of clusters) in 19 out of 20 runs. However, few of the existing validity indices result in the true cluster number.

For DataSet2, the very strong overlapping of clusters and the random initialization procedure result in a very serious consequence: no validity index often yields the true cluster number. Nevertheless, V_{WSJ} produces the correct number of clusters in 8 runs out of 20. Most of the validity indices divide the data set into two clusters, one of which results from the merging of two Gaussian components and the other from the merging of three Gaussian components. We notice that V_{WSJ} and V_{FS} sometime produce 4 clusters. In this case, two clusters are merged and the other three are kept. Compared with others, the results produced by these two validity indices are acceptable.

For DataSet3, it contains 10 clusters and the degree of spread within clusters is different. There are four components tending to merge into two. This type of data set is difficult to identify. V_{Xie}, V_{FS} and V_{WSJ} show the best identification ability for the data, reaching a 90% accuracy rate.

Table 3
The optimal number of clusters by Algo2 for DataSet1

	V_{PC}	V_{PE}	V_{Xie}	V_{FS}	V_{RH}	V_{ZLE}	V_{RLR}	V_{WSJ}
Run1 Run20	2(20)	2(20)	2(20)	3(1),4(3), 5(8), 6(6),8(2)	4(1),5(4), 6(10),7(4),8(1)	2(20)	2(14),3(6)	3(19),5(1)
Accuracy rate	0/20	0/20	0/20	1/20	0/20	0/20	6/20	19/20

Table 4
The optimal number of clusters by Algo2 for DataSet2

	V_{PC}	V_{PE}	V_{Xie}	V_{FS}	V_{RH}	V_{ZLE}	V_{RLR}	V_{WSJ}
Run1 Run20	2(20)	2(20)	2(20)	3(9),4(8), 5(2),6(1)	2(17),3(3)	2(20)	2(20)	2(8), 4(4),5(8)
Accuracy rate	0/20	0/20	0/20	2/20	0/20	0/20	0/20	8/20

Table 5
The optimal number of clusters by Algo2 for DataSet3

	V_{PC}	V_{PE}	V_{Xie}	V_{FS}	V_{RH}	V_{ZLE}	V_{RLR}	V_{WSJ}
Run1 Run20	2(20)	2(20)	7(1),9(1), 10(18)	10(18), 11(1),14(1)	10(8),11(1) 13(2),19(2) 21(5), 23(2)	2(20)	5(19) 7(1)	7(1),9(1), 10(18)
Accuracy rate	0/20	0/20	18/20	18/20	8/20	0/20	0/20	18/20

Table 6
The optimal number of clusters by FBSA for DataSet1

	V_{PC}	V_{PE}	V_{Xie}	V_{FS}	V_{RH}	V_{ZLE}	V_{RLR}	V_{WSJ}
Run1–20	2	2	2	6	7	2	2	3
Correct rate	0/20	0/20	0/20	0/20	0/20	0/20	0/20	20/20

Table 7
The optimal number of clusters by FBSA for DataSet2

	V_{PC}	V_{PE}	V_{Xie}	V_{FS}	V_{RH}	V_{ZLE}	V_{RLR}	V_{WSJ}
Run1–20	2	2	2	4	2	2	2	5
Correct rate	0/20	0/20	0/20	0/20	0/20	0/20	0/20	20/20

Tables 6–8 give the results of the optimal number of clusters obtained by FBSA combined with each validity index for the three data sets. Because the results are the same for each run, we list the first and the last values. For DataSet1 and DataSet2, the optimal number of clusters yielded by V_{WSJ} is the true number of clusters in all 20 runs. However, applying the existing validity indices to FBSA, none obtains the true cluster number. For DataSet3, V_{Xie}, V_{FS}, V_{RH} and V_{WSJ} lead to the true number, 10 clusters. The other validity indices fail to produce the true number.

5.2.2. Error between cluster prototype and component mean

The other criterion we used here was the error between cluster center and component mean. One of the important goals of a clustering algorithm is to find the cluster

Table 8
The optimal number of clusters by FBSA for DataSet3

	V_{PC}	V_{PE}	V_{Xie}	V_{FS}	V_{RH}	V_{ZLE}	V_{RLR}	V_{WSJ}
Run1–20	2	2	10	10	10	2	5	10
Correct rate	0/20	0/20	20/20	20/20	2/20	0/20	0/20	20/20

Table 9
The average error between cluster prototype and component mean by Algo2, for the three data sets

Data	V_{PC}	V_{PE}	V_{Xie}	V_{FS}	V_{RH}	V_{ZLE}	V_{RLR}	V_{WSJ}
DataSet1	0.8016	0.8016	0.8016	2.6661	2.9495	0.8016	0.6732	0.4721
DataSet2	0.5738	0.5738	0.5738	0.5447	0.5528	0.5738	0.5738	0.3990
DataSet3	2.1378	2.1378	1.6892	1.9969	9.1726	2.1378	5.0926	1.6892

Table 10
The average error between cluster prototype and component mean by FBSA, for the three data sets

Data	V_{PC}	V_{PE}	V_{Xie}	V_{FS}	V_{RH}	V_{ZLE}	V_{RLR}	V_{WSJ}
DataSet1	0.8019	0.8019	0.8019	3.2158	3.9313	0.8019	0.8019	0.3853
DataSet2	0.5738	0.5738	0.5738	0.4567	0.5738	0.5738	0.5738	0.2313
DataSet3	2.1356	2.1356	1.6355	1.6355	1.6355	2.1356	5.2356	1.6355

prototypes that represent the component means. For this reason, and since the component means are known for the data sets used, we use the error between cluster prototype and component mean as a criterion. The error $E(C_{Opt})$ is defined as follows: suppose $\{v_i\}(i = 1, 2, \ldots, C_{Opt})$ are the cluster centers and $\{m_i\}(i = 1, 2, \ldots, K)$ the component means (K is the number of components). Then

$$E(C_{Opt}) = \sum_{i=1}^{C_{Opt}} \min_{1 \leqslant j \leqslant K} (\|v_i - m_j\|). \quad (5)$$

We calculated the average values of $E(C_{Opt})$ yielded by Algo2 and FBSA with different validity indices for the three data sets over 20 runs. Table 9 lists the results for Algo2 and Table 10 lists the results for FBSA. According to the explanations of the experimental setting given at the beginning of Section 5.2, if two validity indices always yield the same optimal number of clusters, then they always yield the same clusters. This explains why several validity indices yield the same average value of $E(C_{Opt})$ for a data set.

From the two tables, we note that the results for FBSA and Algo2 are quite similar. FBSA is slightly better than Algo2 for DataSet3 and slightly worse for DataSet1. Both algorithms perform very well when the validity index V_{WSJ} is used. At least for these three data sets, FBSA combined with V_{WSJ} seems to be the best combination. The reason for the good performance displayed by both algorithms when combined with V_{WSJ} is the accurate estimation of the number of clusters. This corroborates the results of the previous subsection. The most important conclusion suggested by these experiments is that the accuracy of the new algorithm FBSA does not suffer from the restricted initialization scheme, while it is much more time-efficient, as we will show in Section 5.4.

5.3. Stability across different runs

An interesting property of FBSA is its stability across different runs, which can be observed from the tests on the three data sets. The output of these experiments (Tables 6–8) is independent of the initial cluster centers, whereas when Algo2 is applied to each of these data sets, the output varies quite significantly depending on the validity index used (Tables 3–5). The way in which new cluster centers are initialized at each phase of FBSA is certainly the reason for its stability. Since FBSA performs at least as well as Algo2 in computing the number of clusters and the cluster centers, this property of stability could make it a more interesting choice than Algo2 in many practical applications.

5.4. Comparison of FBSA to Algo2 in Terms of Time Cost

Here we will show the performance of the new algorithms by comparing their numbers of iterations and the real run time needed for convergence. Algo2 and FBSA both use the same validity index, V_{WSJ}. In comparing FBSA with Algo2,

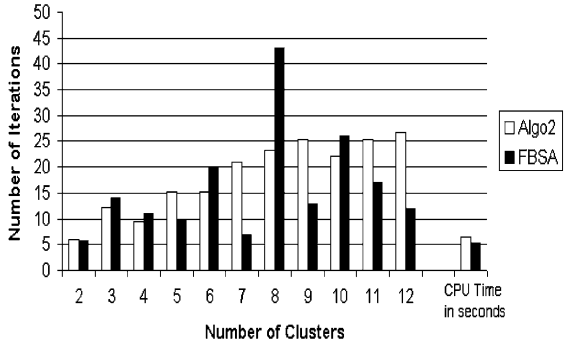

Fig. 5. Comparison of the number of iterations and run time on DataSet1 for Algo2 and FBSA.

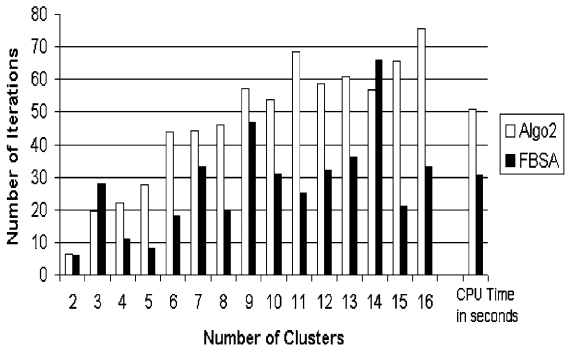

Fig. 6. Comparison of the number of iterations and run time on DataSet2 for Algo2 and FBSA.

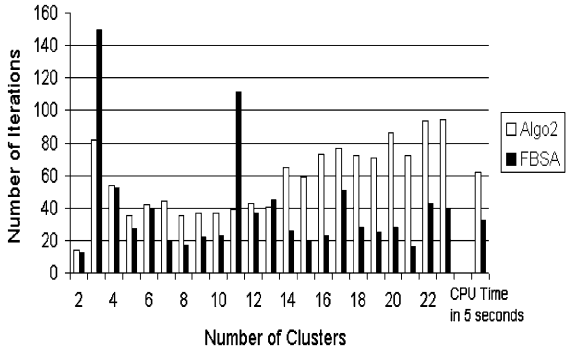

Fig. 7. Comparison of the number of iterations and run time on DataSet3 for Algo2 and FBSA.

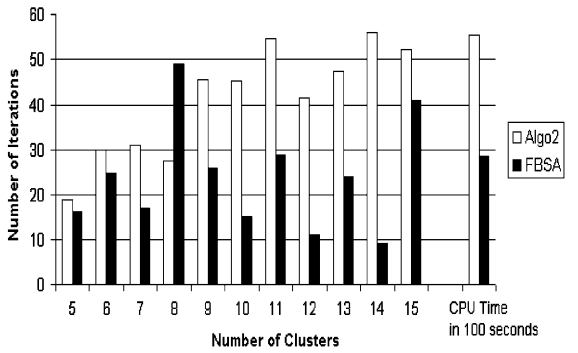

Fig. 8. Comparison of the number of iterations and run time on DataSet4 for Algo2 and FBSA.

we are interested in both the reduction in the number of iterations for each subsequent value of c tested and the run time. We ran this experiment on a PC with Pentium III 450 MHz CPU and 320 MB RAM.

5.4.1. Test on DataSet1, DataSet2 and DataSet3

Figs. 5–7 show the numbers of iterations that Algo2 and FBSA need for each value of c and their run times. In all cases, the new algorithm requires less run time to converge than Algo2. The improvement in the run times is obvious. Although there are some variations in the number of iterations for some c, we can see that the reduction in the number of iterations relative to Algo2 is significant for all data sets, especially when c is large.

5.4.2. Test on DataSet4

In addition to the data sets described above, we also tested a larger data set with 60,000 data vectors in order to get a better idea about the improvement in speed. For statistical purposes, we ran each algorithm 10 times and recorded the average number of iterations for each c and the CPU time. We ran this experiment on a PC computer with a 2.40 GHz CPU and 1 Gb RAM.

DataSet4 originated from a file provided by Statistics Canada under its Data Liberalization Initiative Program (http://www.lib.unb.ca/gddm/data/Ftp_famex.html). The FAMEX file from StatsCan is a survey on household expenditures and budgets for the year 1996. It includes expenditures, income, and changes in assets and debts. The variables include composition of household, characteristics of dwelling, shelter expenses, food and alcohol, clothing, medical and health care, travel and transportation, recreation and education and tobacco. After a simple preprocessing (removing non-numeric variables and items with missing values), we obtained a data set with 22 variables and 600,000 data vectors. DataSet4 is a randomly selected subset of 60,000 items.

The main objective of testing DataSet4 was to further illustrate the computational efficiency of the proposed algorithm. For a real data set, the actual number of clusters is often unknown. Subjective evaluation of the clustering results is beyond the scope of this paper. We will restrict ourselves to the evaluation of time efficiency. Fig. 8 shows the number of FCM iterations needed as a function of c, the number of clusters tested. In the same figure, we also show

the comparison of the real CPU time. For the number of iterations, we obtained a similar profile as in the experiments conducted on the three previous data sets. Also, in terms of gain in CPU time, FBSA is 48.15% faster than Algo2, which is similar to the gain obtained in the previous examples. Thus FBSA seems to scale well to a large data set.

5.5. Discussion

From these results and from the results we obtained on similar data sets, we can draw the following conclusions:

(1) The new validity index proposed here significantly improves the performance in determining the number of clusters and the accuracy of cluster centers. In fact, based on the three sets of experimental results, only our new validity index is able to yield the correct number of clusters consistently, whether Algo2 or FBSA is used. Comparison of the cluster centers and the component means shows that FBSA combined with V_{WSJ} gives accurate prototypes.
(2) The cluster centers yielded by FBSA are accurate for small data sets containing fewer than several hundred data (which is the case for most public domain data sets), even when there are overlapping clusters. We also obtained similar results on large data sets generated from a mixture of Gaussians. However, it is difficult to carry out these comparisons on large real-world data sets because the information required for the comparison, such as the true number of clusters and the true cluster centers, is often not available.
(3) In terms of computational efficiency, FBSA generally needs less run time than Algo2. For some c, the number of iterations for FBSA is larger than for Algo2. In these cases, c has exceeded the true number of clusters: obviously more iterations are needed to force partition of a data set into more clusters than it really has.
(4) It is easy to understand that the number of iterations for each value of c tends to be small when c is close to the true number of clusters. When the value of c moves away from the true number of clusters, the algorithm FBSA tends to exhibit a more abrupt increase in the number of iterations. Another interesting property of FBSA is that the number of iterations for different runs is very stable for each value of c. This is illustrated in Tables 6–8. These properties can be useful for developing a strategy to further limit the search range of c.

6. Conclusion and perspectives

The major contributions of this paper are an improved FCM-based algorithm for determining the number of clusters and a new index for validating clustering results. The new validity index, V_{WSJ}, is a function of the original data, cluster centers and membership. Experimental results have shown that the new index is able to yield accurate numbers of clusters even for data sets with overlapping clusters, where existing indices often display unpredictable behavior. The new improved clustering algorithm has shown advantages in terms of computation time and stability, compared with the basic trial-and-error FCM-based algorithms. All experiments show that the combination of FBSA and V_{WSJ} gets the best results. It may also be more useful in dealing with large high-dimensional data sets. In this case, it is possible to test for big jumps in the value of $V_{WSJ}(c)$ or in the number of iterations in order to halt the clustering procedure before testing larger numbers of clusters. Investigation into the conditions for stopping FBSA is under way.

There are a number of other avenues for further investigation. First, an extensive evaluation of the new index and existing indices in terms of sensitivity to cluster overlapping is much needed for data-mining applications. Such an evaluation can only be carried out with generated data sets, since they allow us to test clustering algorithms in a more controlled way. In Refs. [19,20], Wang et al. have proposed a formal definition of the overlapping rate between clusters and have developed methods for automatically generating data sets. These methods are being extended to generate realistic high-dimensional data sets with outliers and will be used to evaluate the index and algorithm proposed in this paper.

Second, it is necessary to carry out theoretical studies on why the new index V_{WSJ} is able to yield such satisfactory results. To our knowledge, all past work dealing with the evaluation of validity indices has used experimental results to illustrate the performance of an index. Combining concepts and techniques used in statistics with appropriate data models could provide some insight into the behavior of validity indices.

Finally, investigation into the use of the new algorithm to deal with the dimension reduction problem is another promising avenue. For instance, selection of appropriate dimensions (features) for a supervised classification problem can be performed by successively applying the clustering algorithm to different combinations of the feature dimensions. The clustering results can be evaluated based on different criteria to select the best combination.

Acknowledgements

This work has been supported by research grants from the Natural Sciences and Engineering Research Council of Canada (NSERC) to Dr. Shengrui Wang. The work has also been supported in part by a grant from Network Centers of Excellence MITACS and AUTO21 to Dr. Shengrui Wang. We would also thank the anonymous referees for their valuable comments.

References

[1] J. Han, M. Kamber, Datamining: Concepts and Techniques, Morgan Kaufmann Publishers, San Francisco, 2001.

[2] F. Hoppner et al., Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition, Wiley, New York, 1999.

[3] R. Dubes, Cluster analysis and related issues, Handbook of Pattern Recognition and Computer Vision, World Scientific Publishing Co., Inc., River Edger, NJ, USA, 1993, pp. 3–32.

[4] B.S. Everitt, Cluster Analysis, 3rd Edition, Edward Arnold, London, 1993.

[5] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Englewood Cliffs, NJ, 1988.

[6] A. Abeantes, J. Marques, A class of constrained clustering for object boundary extraction, IEEE Trans. Image Process. 5 (1996) 1507–1521.

[7] R. Krishnapuram, O. Nasraouui, J. Keller, The fuzzy C-spherical shells algorithm: a new approach, IEEE Trans. Neural Networks 3 (5) (1992) 663–671.

[8] X. Xie, G. Beni, A validity measure for fuzzy clustering, IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) 13 (8) (1991) 841–847.

[9] Y. Fukuyama, M. Sugeno, A new method of choosing the number of clusters for the fuzzy C-means method, in: Proceedings of Fifth Fuzzy Systems Symposium, 1989, pp. 247–250.

[10] M. Rezae, B. Letlieveldt, J. Reiber, A new cluster validity index for the fuzzy c-means, Pattern Recogn. Lett. 19 (1998) 237–246.

[11] H. Rhee, K. Oh, A validity measure for fuzzy clustering and its use in selecting optimal number of clusters, in: Proc. IEEE, 1996, pp. 1020–1025.

[12] A. Baraldi, P. Blonda, A survey of fuzzy clustering algorithms for pattern recognition—part I, IEEE Trans. Syst. Man, Cybern. SMC-29 (1999) 778–785.

[13] J. Pena, J. Lozano, P. Larranaga, An empirical comparison of four initialization methods for the k-means algorithm, Pattern Recogn. Lett. 20 (1999) 1027–1040.

[14] T. Gonzalez, Clustering to minimize and maximum intercluster distance, Theor. Comput. Sci. 38 (1985) 293–306.

[15] J.C. Bezdek, Pattern Recognition in Handbook of Fuzzy Computation. IOP Publishing Ltd., Boston, Ny, 1998 (Chapter F6).

[16] M.L.A.E.N. Zahid, O. Abouelala, Unsupervised fuzzy clustering, Pattern Recognition Lett. 20 (1999) 123–129.

[17] N. Pal, J. Bezdek, On cluster validity for the fuzzy C-means model, IEEE Trans. Fuzzy Systems 3 (3) (1995) 370–390.

[18] S. Halgamuge, M. Glesner, Neural networks in designing fuzzy systems for real world applications, Fuzzy Sets Systems 65 (1) (1994) 1–12.

[19] E. Aitnouri, S. Wang, D. Ziou, On clustering techniques comparison for histogram pdf estimation, J. Pattern Recognition Image Anal. 10 (2) (2000) 206–217.

[20] E.M. Aitnouri, F. Dubeau, S. Wang, D. Ziou, Controlling mixture component overlap for clustering algorithms evaluation, Pattern Recognition and Image Analysis 12 (4) (2002) 331–346.

About the Author—HAOJUN SUN, Ph.D. Student at the Department of Computer Science of the University of Sherbrooke, Canada. He was associate professor from 1998 to 2000 and assistant professor from 1988 to 1998 in computer science, Hebei University, China. His research interests include data mining, cluster analysis, information retrieval, fuzzy set and system and pattern recognition.

About the Author—SHENGRUI WANG, Professor at the Department of Computer Science of the University of Sherbrooke. He received his B.S. degree in mathematics from Hebei University, China, in 1982, M.S. degree in applied mathematics from the Université de Grenoble in 1986 and Ph.D. degree from the Institut National Polytechnique de Grenoble, France, in 1989. During 1990, he worked as a post-doc fellow at the Department of Electrical Engineering at the Laval University. His research interests include pattern recognition, neural networks, data mining, image processing, image database, geographical information system and navigation systems.

About the Author—QINGSHAN JIANG, Professor at the Department of Computer Science of Xiamen University, China. He received a Ph.D. in mathematics from Chiba Institute of Technology, Japan, in 1996, and a Ph.D. in computer science from University of Sherbrooke, Canada, in 2002. During his over 20 years of study and research, he has published over 30 scientific papers at international journals and conference proceedings. His expertise lies in system development with a strong focus in the areas of image processing, statistical analysis, fuzzy modelling and data mining.