Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Pattern Recognition Letters 25 (2004) 1123–1132

Pattern Recognition Letters

www.elsevier.com/locate/patrec

Improving fuzzy *c*-means clustering based on feature-weight learning

Xizhao Wang [a,*], Yadong Wang [b], Lijuan Wang [a,b]

[a] *Department of Mathematics and Computer Science, Hebei University, Baoding, Hebei 071002, China*
[b] *Department of Computer Science and Engineering, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China*

Received 11 June 2003; received in revised form 12 January 2004
Available online 22 April 2004

Abstract

Feature-weight assignment can be regarded as a generalization of feature selection. That is, if all values of feature-weights are either 1 or 0, feature-weight assignment degenerates to the special case of feature selection. Generally speaking, a number in $[0,1]$ can be assigned to a feature for indicating the importance of the feature. This paper shows that an appropriate assignment of feature-weight can improve the performance of fuzzy *c*-means clustering. The weight assignment is given by learning according to the gradient descent technique. Experiments on some UCI databases demonstrate the improvement of performance of fuzzy *c*-means clustering.

Keywords: Fuzzy *c*-means; Weighted fuzzy *c*-means; Fuzziness; Similarity measure; Gradient descent technique

1. Introduction

Clustering analysis, which leads to a crisp or fuzzy partition of sample space, has been widely used in a variety of areas such as data mining and pattern recognition [e.g. Hall et al., 1992; Cannon et al., 1986; Bezdek, 1981]. Fuzzy *c*-means (FCM) proposed by Dunn (1974) and extended by Bezdek (1981) is one of the most well-known methodologies in clustering analysis.

Basically FCM clustering is dependent of the measure of distance between samples. In most situations, FCM uses the common Euclidean distance which supposes that each feature has equal importance in FCM. This assumption seriously affects the performance of FCM, since in most real world problems, features are not considered to be equally important. For example, we consider Iris database (Fisher, 1936) which has four features, i.e., sepal length (SL), sepal width (SW), petal length (PL) and petal width (PW). Fig. 1 shows a clustering for Iris database based on features SL and SW, while Fig. 2 shows a clustering based on PL and PW. From Fig. 1, one can see that there are much more crossover between the star class and the point class. It is difficult for us to discriminate the star class from the point class. On the other hand, it is easy to see that Fig. 2 is more

[*] Corresponding author. Fax: +86-312-507-9638.
E-mail address: wangxz@mail.hbu.edu.cn (X. Wang).

0167-8655/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2004.03.008

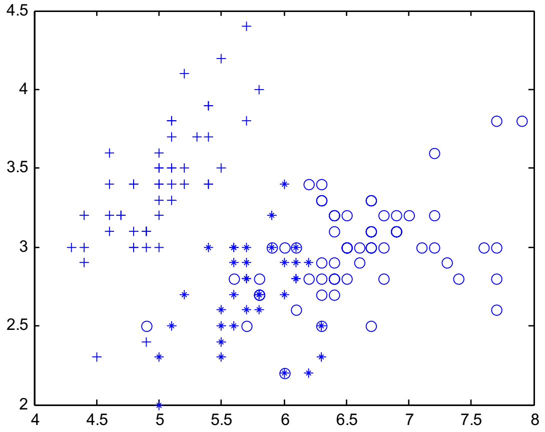

Fig. 1. Clustering result of Iris database based on feature-weights $(1,1,0,0)$ by FCM algorithm.

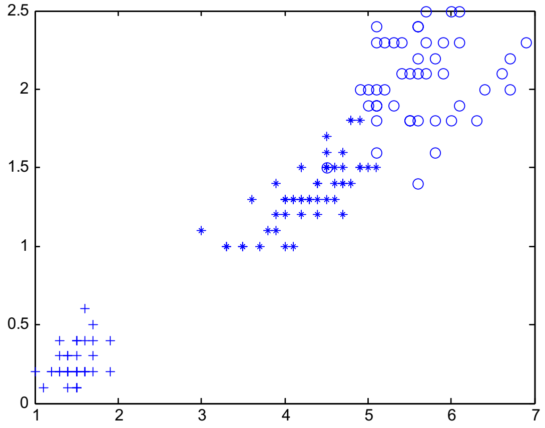

Fig. 2. Clustering result of Iris database by FCM based on feature-weights $(0,0,1,1)$.

crisp than Fig. 1. It illustrates that, for the classification of Iris database, features PL and PW are more important than SL and SW. Here we can think of that the weight assignment $(0,0,1,1)$ is better than $(1,1,0,0)$ for Iris database classification.

FCM clustering is sensitive to the selection of distance metric. In (Zhao, 1987), the author stated that the Euclidean distance can give good results when all clusters are spheroids with same size or when all clusters are well separated. In (Gustafson and Kessel, 1979; Krishnapuram and Kim, 1999), the authors proposed a G–K algorithm which uses the well-known Mahalanobis distance as the metric in FCM. They reported that the G–K algorithm is better than Euclidean distance based algorithms when the shape of data is considered. In (Wu and Yang, 2002), the authors proposed a new robust metric, which is distinguished from the Euclidean distance, to improve the robustness of FCM.

Since FCM's performance depends on selected metrics, it will depend on the feature-weights which are incorporated into the Euclidean distance. We try in this paper to adjust these feature-weights to improve FCM's performance.

Each feature is considered to have an importance degree which is called feature-weight. Feature-weight assignment is an extension of feature selection. The latter has only either 0-weight or 1-weight value, while the former can have weight values in the interval $[0,1]$. From existing literatures one can find a number of commonly used feature selection algorithms including of sequential unsupervised feature selection algorithms (Dash and Liu, 2000), wrapper approaches based on expectation maximization (Dy and Brodely, 2000), maximum entropy based methods (Basu et al., 2000), GA based methods (Pal and Wang, 1996), and neuro-fuzzy approaches (Pal et al., 2000). Generally speaking, feature selection method cannot be used as feature-weight learning technique, but the inverse is right. This paper proposes an approach to feature-weight learning which is based on the gradient-desent technique. It shows that an appropriate assignment to feature-weights can improve the performance of FCM clustering. Experiments on some UCI databases demonstrate the improvement of performance of FCM clustering.

This paper has following organization. Section 2 first reviews the FCM clustering algorithm. And then lists some cluster validity functions which are used to measure the performance of FCM. Section 3 introduces a feature-weight learning algorithm based on the gradient-decent technique. Section 4 experimentally demonstrates the improvement of performance of FCM on some UCI databases. And the final Section 5 offers our conclusion.

2. FCM algorithm and its some validity functions

2.1. FCM algorithm

FCM partitions a set of s-dimensional vectors $X = \{X_1, \dots, X_n\}$ into c clusters where $X_j = \{x_{j1}, \dots, x_{js}\}$ represents the jth sample for $j = 1, \dots, n$. Every cluster is a fuzzy set defined on the sample space $X = \{X_1, \dots, X_n\}$. The ith cluster is supposed to have the center vector $v_i = \{v_{i1}, \dots, v_{is}\}$ $(1 \leqslant i \leqslant c)$. FCM can be regarded as an extension of HCM (the Hard (i.e., crisp) c-means). The main difference between FCM and HCM is that the generated partition is fuzzy for FCM and is crisp for HCM. For the jth sample X_j $(1 \leqslant j \leqslant n)$ and the ith cluster center v_i $(1 \leqslant i \leqslant c)$, there is a membership degree u_{ij} $(\in [0, 1])$ indicating with what degree the sample X_j belongs to the cluster center vector v_i, which results in a fuzzy partition matrix $U = (u_{ji})_{n \times c}$. FCM aims to determine cluster centers v_i $(i = 1, 2, \dots, c)$ and the fuzzy partition matrix U by minimizing the objective function J defined as follows:

$$J(U, v_1, v_2, \dots, v_c; X) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m d_{ij}^2 \tag{2.1}$$

subject to

$$\sum_{i=1}^{c} u_{ij} = 1, \quad \forall j = 1, \dots, n \tag{2.2}$$

where d_{ij} is Euclidean distance from sample X_j to cluster center v_i defined as:

$$d_{ij} = \sqrt{\sum_{k=1}^{s} (v_{ik} - x_{jk})^2} \tag{2.1a}$$

The exponent m in Eq. (2.1) is used to adjust the weighting effect of membership values. Large m will increase the fuzziness of the function (2.1). The value of m is often set to 2. Applying derivative to Eqs. (2.1) and (2.2), one can derive the computational formulae of u_{ij} and v_i as:

$$v_i = \frac{\sum_{j=1}^{n} u_{ij}^m X_j}{\sum_{j=1}^{n} u_{ij}^m} \tag{2.3}$$

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}} \right)^{2/(m-1)}} \quad (m \neq 1) \tag{2.4}$$

where X_j represents the jth sample.

Based on Eqs. ((2.1)–(2.4)), we describe the FCM algorithm as follows:

Step 1: Choose an integer c and a threshold value ε. Let $m = 2$. Initialize the fuzzy partition matrix U by generating $c \times n$ random numbers in the interval $[0, 1]$.

Step 2: Compute v_i $(1 \leqslant i \leqslant c)$ according to Eq. (2.3).

Step 3: Compute all d_{ij} according to (2.1a) and then all u_{ij} according to (2.4). Thus update the fuzzy partition matrix U by the new computed u_{ij}.

Step 4: Compute the objective function J by using (2.1). If it converges or the difference between two adjacent computed values of objective function J is less than the given threshold ε then stop. Otherwise go to step 2.

The input to FCM algorithm is a set of samples $\{X_1, \dots, X_n\}$ and during executing the algorithm two parameters (m and ε) need to be given in advance. Moreover, the number of clusters is also required to predefine. The output of FCM algorithm is those cluster-centers and the fuzzy partition matrix U.

2.2. Cluster validity functions

Cluster validity functions are often used to evaluate the performance of clustering in different indexes and even two different clustering methods. A lot of cluster validity criteria were proposed during the last 10 years. Most of them came from different studies dealing with the number of clusters. Among the criteria (Dubes and Jain, 1988), there are two important types for FCM. One is based on the fuzzy partition of sample set and the other is on the geometric structure of sample set.

The main idea of validity functions based on fuzzy partition is that, the less fuzziness of the partition is, the better the performance is. The

Table 1
A brief summary of four selected validity functions

Validity function	Functional description	Optimal partition
Partition coefficient	$V_{\mathrm{pc}}(U)=\frac{\sum_{j=1}^{n}\sum_{i=1}^{c}u_{ij}^2}{n}$	$\mathrm{Max}(V_{\mathrm{pc}})$
Partition entropy	$V_{\mathrm{pe}}(U)=-\frac{1}{n}\left\{\sum_{j=1}^{n}\sum_{i=1}^{c}\left[u_{ij}\log u_{ij}\right]\right\}$	$\mathrm{Min}(V_{\mathrm{pe}})$
Fukuyama–Sugeno function	$V_{\mathrm{fs}}(U,v_1,\mathrm{L},v_c;X)=\sum_{i=1}^{c}\sum_{j=1}^{n}u_{ij}^2\left(\Vert X_j-v_i\Vert^2-\Vert v_i-\bar{v}\Vert^2\right)$	$\mathrm{Min}(V_{\mathrm{fs}})$
Xie–Beni function	$V_{\mathrm{xb}}(U,v_1,\mathrm{L},v_c;X)=\frac{\sum_{i=1}^{c}\sum_{j=1}^{n}u_{ij}^2\Vert X_j-v_i\Vert^2}{n*\left(\min_{i\neq k}\left\{\Vert v_i-v_k\Vert^2\right\}\right)}$	$\mathrm{Min}(V_{\mathrm{xb}})$

representative functions for this type are partition coefficient V_{pc} (Bezdek, 1974) and partition entropy V_{pe} (Bezdek, 1975). Empirical studies [e.g. Dunn, 1976; Bezdek et al., 1980] think of that the maximum V_{pc} and minimum V_{pe} may be lead to a good interpretation of the samples considered. The best interpretation is achieved when the value V_{pc} gets its maximum or V_{pe} gets its minimum. Disadvantage of V_{pc} and V_{pe} are the lack of direct connection to a geometrical property and its monotonic decreasing tendency with c. On the other hand the main idea of the validity function based on the geometric sample structure is that samples within one partition should be compact and samples within different clusters should be separate, for instance, Fukuyama–Sugeno function V_{fs} (Fukuyama and Sugeno, 1989) and the Xie–Beni function V_{xb} (Xie and Beni, 1991). It is clear that a good partition is that the samples in one cluster are compact and the samples among different clusters are separate. Minimizing V_{xb} or minimizing V_{fs} is expected to lead to a good partition. Because V_{xb} decreases monotonically when c gets very large, we can impose a punishing function. But V_{xb} can get a better performance even without a punishing function (Xie and Beni, 1991). Intuitively, the fuzziness and the compactness of a partition should decrease with the increase of the number of clusters. For example, the partition entropy decreases to zero when c gets very large and goes to n.

It is not possible to compare two crisp partitions with the validity functions based on the fuzziness of the partition, since the fuzziness of any crisp partition is zero. For example, the first two validity functions, partition coefficient and partition entropy, will be a constant when all membership degrees are either 0 or 1. It is hard to generally say that a good partition is crisp or fuzzy. It depends on the specific data set and its explanation. However, if we consider all fuzzy partitions (including of the crisp ones) for a data set, the good partition should meet that (A) the objective function converges, and (B) its fuzziness is as small as possible.

Table 1 is a brief summary of 4 selected cluster validity functions which will be used in Section 4 to evaluate the performance of FCM clustering. It is noted that in some cases these validity functions cannot get their optimal values simultaneously.

3. Feature-weight learning

The feature-weight learning is based on the similarity between samples. There are many ways to define the similarity measure, such as the related coefficient and Euclidean distance, etc. Motivated by simplicity and easy-manipulation, here the similarity measure $\rho_{ij}^{(w)}$ is defined as follows:

$$\rho_{ij}^{(w)}=\frac{1}{1+\beta*d_{ij}^{(w)}}\tag{3.1}$$

Since similarity measure (3.1) is associated with the weighted Euclidean distance, it has well analytic properties and intuitive meaning. The value of similarity measure $\rho_{ij}^{(w)}$ is called new similarity degree. When $w=(1,1,\dots,1)$ the similarity degree $\rho_{ij}^{(1)}$ (in short ρ_{ij}) is called old. We hope ρ_{ij} can uniformly distribute in $[0,1]$. However most real data sets may not meet the requirement of uniform distribution in $[0,1]$. To adjust the mean of the distribution of $\rho_{ij}s$, the positive parameter β (>0) is used. Noting that 0.5 is the mean of the uniform distribution in $[0,1]$, we would like to select a β such that:

$$\frac{2}{n(n-1)}\sum_{j<i}\frac{1}{1+\beta * d_{ij}}=0.5 \tag{3.1a}$$

where d_{ij} is commonly used Euclidean distance, and $d_{ij}^{(w)}$ is the weighted Euclidean distance defined as follows:

$$d_{ij}^{(w)}=\sqrt{\left(\sum_{k=1,\ldots,s}w_k^2(x_{ik}-x_{jk})^2\right)} \tag{3.1b}$$

where $w=(w_1,\ldots,w_s)$ is the feature-weight vector. Its component is the importance degree corresponding to each feature. Larger w_k is, more important the kth feature is in FCM. When $w=(1,\ldots,1)$, the space $\{\|d_{ij}^{(w)}\|\leqslant r\}$ is a hypersphere with radius r in the well-known Euclidean space (called original space). In the original space, $d_{ij}^{(w)}$ is denoted by d_{ij} and $\rho_{ij}^{(w)}$ by ρ_{ij}. When $w\neq(1,\ldots,1)$, it means that the axes would be extended or shrunk in accordance with w_k. Thus the space $\{\|d_{ij}^{(w)}\|\leqslant r\}$ is hyper-ellipse, called the transformed space. The lower the value of w_k is, the higher the flattening extent is.

According to De Luca and Termini (1972), the fuzziness of similarity degrees $\{\rho_{ij}|i<j\}$ can be defined as

$$\text{Fuzziness}=\frac{-2}{n(n-1)}\sum_{i<j}\big(\rho_{ij}\log\rho_{ij} + (1-\rho_{ij})\log(1-\rho_{ij})\big) \tag{3.1c}$$

It is clear that fuzziness shown in (3.1c) attains its maximum when all similarity degrees are close to 0.5. It will attains its minimum when all similarity degrees are close to either 0 or 1. A good partition should have the following property: the samples within one cluster are closed to the center and different centers are more separate, which implies that the samples within one cluster are more similar ($\rho_{ij}^{(w)}\to 1$) and dissimilar samples are more separate ($\rho_{ij}^{(w)}\to 0$), so that the fuzziness given in (3.1c) is low. Therefore we hope that, by adjusting w, the similar objects ($\rho_{ij}>0.5$) in the original space are more similar ($\rho_{ij}^{(w)}\to 1$) in the transformed space, and the dissimilar objects ($\rho_{ij}<0.5$) in the original space are more separate ($\rho_{ij}^{(w)}\to 0$) in the transformed space. It is expected to lead an improvement of FCM's performance.

Based on the above discussion, we learn the feature-weight value by minimizing an evaluation function $E(w)$ which was first introduced in (Basak et al., 1998) and then was applied to clustering performance improvement (Yeung and Wang, 2002). $E(w)$ is defined as follows:

$$E(w)=\frac{2}{n(n-1)}\times\sum_i\sum_{j\neq i}\frac{1}{2}\left(\rho_{ij}^{(w)}(1-\rho_{ij})+\rho_{ij}\big(1-\rho_{ij}^{(w)}\big)\right) \tag{3.2}$$

We can use the gradient descent technique to minimize $E(w)$. Let Δw_k be the change of w_k, compute as follows:

$$\Delta w_k=-\eta\frac{\partial E(w)}{\partial w_k}\ (0<k<s) \tag{3.3}$$

For the procedure and related details, one can refer to Yeung and Wang (2002).

After obtaining feature-weight values by above learning, we can use the weighted Euclidean distance to replace the common Euclidean distance in FCM. In this way, the objective function $J^{(w)}$ given in Eq. (2.1) will become the following:

$$J^{(w)}(U,v_1,\ldots,v_c;X)=\sum_{i=1}^{c}\sum_{j=1}^{n}u_{ij}^m(d_{ij}^{(w)})^2 \tag{3.4}$$

Minimizing (3.4) subject to (2.2), we then can obtain u_{ij} and v_i as follows

$$v_i=\frac{\sum_{j=1}^n u_{ij}^m x_j}{\sum_{j=1}^n u_{ij}^m} \tag{3.5}$$

$$u_{ij}=\frac{1}{\sum_{k=1}^c\left(\frac{d_{ij}^{(w)}}{d_{kj}^{(w)}}\right)^{2/(m-1)}}\quad(m\neq 1) \tag{3.6}$$

The other parts of the algorithm are same as FCM given in Section 2.

For the weighted distance with learned feature-weight values, the FCM algorithm described in Section 2 is still available. We call the FCM with learned feature-weight values weighted FCM, in short, WFCM. The only difference between FCM and WFCM is the different distance metrics. Like the FCM, WFCM algorithms also input all samples and output the cluster centers and the partition matrix $U^{(w)}$.

From existing references such as Gustafson and Kessel (1979) and Krishnapuram and Kim (1999), some works related to weighted fuzzy c-mean algorithm can be found. Comparing our proposed WFCM with G–K algorithm and AFC proposed in (Gustafson and Kessel, 1979; Krishnapuram and Kim, 1999), we find the following similarity and difference.

Similarity:

(A) Three clustering algorithms are of FCM type, minimizing an objective function of the type

$$J(B,U,X) = \sum_{i=1}^{c}\sum_{j=1}^{b}(u_{ij})^m d^2(x_j,m_i) = \sum_{i=1}^{c}\sum_{j=1}^{b}(u_{ij})^m \left(x_j-m_i\right)^{\mathrm{T}}A_i\left(x_j-m_i\right)$$

where A_i is a diagonal and positive definite matrix and other symbols have the same meaning as reference (Krishnapuram and Kim, 1999).

(B) The three algorithms use, directly or indirectly, the weighted Euclidean distance

$$d^2(x_j,m_i) = \left(x_j-m_i\right)^{\mathrm{T}}A_i\left(x_j-m_i\right).$$

Difference:

(A) The weights are fixed in our proposed WFCM, but are variant for different clusters in G–K and AFC.

(B) In WFCM, the weights are learned from minimizing Eq. (3.2). But in G–K and AFC, the weights (A_is) are learned from minimizing the objective function.

(C) Minimizing the objective function may lead to such a situation that two objects are very similar (dissimilar, resp.) in Euclidean space (the original metric space) but dissimilar (very similar, resp.) in weighted Euclidean space (the transformed metric space). It is an obvious defect. Eq. (3.2) is designed in order to overcome this defect.

(D) $|A_i|$ is fixed a priori in G–K, and is estimated from data in AFC. But in our proposed WFCM, this problem does not exist.

(E) G–K and AFC are derived based on the covariance matrix of each cluster and, therefore, both are suitable for well-distributed data sets. WFCM is derived based on the weighted Euclidean distance.

A detailed comparative study on the three algorithms will lead to the very complicated equation derivation and algorithm implementation but will be very interesting. We will later separately report comparative results of the performance for the three algorithms.

4. Experimental demonstration

In this section, we would like to experimentally demonstrate the improvement of performance by comparing the FCM and WFCM. Here, the performance of clustering is measured by the four validity functions listed in Section 2.

Example 1. Let CL be a set of vectors $\mathrm{CL} = \{X_1,X_2,X_3,X_4,X_5\}$, where $X_1 = \{4.8,5.0,3.0,2.0\}$, $X_2 = \{2.0,3.0,4.0,5.0\}$, $X_3 = \{5.0,5.0,2.0,3.0\}$, $X_4 = \{1.0,5.0,3.0,1.0\}$, $X_5 = \{1.0,4.9,5.0,1.0\}$. The FCM clustering result is shown in Table 2 and the WFCM clustering result is shown in Table 3. The performance is evaluated by four evaluated indexes given in Section 3. The number of clusters ranges from 2 to 4, from which we can choose the clustering with best number of clusters, i.e., the clustering with optimal validity function values.

We can draw a conclusion that the performance of Table 3 is generally better than that of Table 2.

Table 2
The FCM clustering result

Number of clusters	The value of validity functions			
	V_{pc}	V_{pe}	V_{xb}	V_{fs}
2	0.81	0.45	0.16	−3.66
3	0.90	0.34	0.02	−26.66
4	0.94	0.18	0.36	−32.02

Table 3
The WFCM clustering result with weight $(0.488, 1, 0, 0)$

Number of clusters	The value of validity functions			
	V_{pc}	V_{pe}	V_{xb}	V_{fs}
2	0.87	0.31	0.027	−15.00
3	0.99	0.01	0.00009	−32.39
4	0.91	0.23	0.002	−30.24

For both 2 and 3 clusters, the WFCM is better than FCM for the mentioned 4 indexes. One may use the 4 index values to determine the number of clusters. The optimal index values correspond to the best numbers of clusters. The best number of clusters is 4 in Table 2 and is 3 in Table 3. The latter is much better than the former. So we think the samples should be clustered into 3 clusters. It is noted that the optimal values of the 4 indexes may not be attained simultaneously. In this situation, a fused result for the 4 indexes may be used to determine the best number of clusters.

Example 2. Compare the clustering result of Iris database by (1) FCM, (2) FCM based on features SL and SW, (3) FCM based on features PL and PW and (4) WFCM. Because Iris database has four features, the FCM clustering result cannot be shown in graph. FCM clustering based on features SL and SW of Iris database is shown in Fig. 1, and FCM clustering based on PL and PW is shown in Fig. 2. The learned feature-weight vector is $(0.0001, 0.0002, 1.0, 0.164)$. Due to the first two features-weight values too small, we omit them in the graph and in this way the WFCM clustering result is shown in Fig. 3. The error rate for the four clustering result is shown in Table 4. The values of evaluation functions shown in Table 5 are used to evaluate the clustering performance. We have the following conclusions:

(1) From Table 4, we can see the error rate for FCM is 15/150. The performance of FCM clustering based on features SL and SW is the worst case whose error rate is 28/150. On the other hand FCM clustering based on features PL and PW and WFCM have a better performance than that of FCM. They have the same error rate: 8/150. Therefore we think of that features PL and PW for Iris database are more important (than other two features) in clustering and FCM clustering based on features PL and PW and WFCM can improve FCM clustering performance.

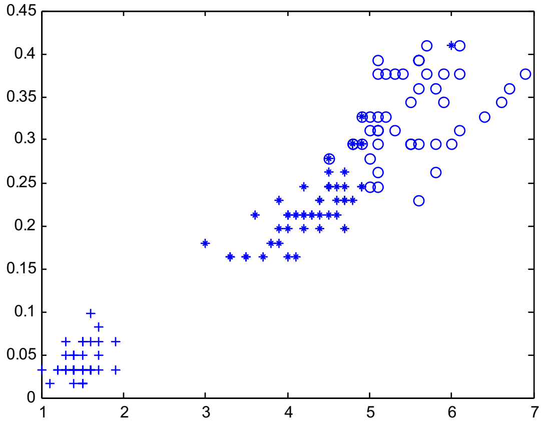

Fig. 3. Clustering of Iris database by FCM based on feature-weights $(0.0001, 0.0002, 1.0, 0.164)$.

Table 4
The error rate for clustering of Iris database by FCM, FCM based on features SL and SW, FCM based on features PL and PW and WFCM

Feature name	Feature-weight	Error rate
(SL, SW, PL, PW)	$1, 1, 1, 1$	15/150
(SL, SW)	$1, 1, 0, 0$	28/150
(PL, PW)	$0, 0, 1, 1$	8/150
(SL, SW, PL, PW)	$0.0001, 0.0002, 1.0, 0.164$	8/150

Table 5
Compare clustering results of Iris database by FCM, FCM based on features SL and SW, FCM based on features PL and PW, and WFCM

Feature-weight	The value of validity functions			
	V_{pc}	V_{pe}	V_{xb}	V_{fs}
$(1, 1, 1, 1)$	0.78	0.57	0.13	−442.94
$(1, 1, 0, 0)$	0.70	0.78	0.18	−353
$(0, 0, 1, 1)$	0.86	0.38	0.06	−518.85
$(0.0001, 0.0002, 1.0, 0.164)$	0.86	0.36	0.05	−529.11

(2) It is easy to see that Figs. 2 and 3 are more crisp than Fig. 1. We can get the same conclusion as that in (1).

(3) From four evaluation function values in Table 5, it can be found out that the performance of WFCM is slightly better than that of FCM clustering based on features PL and PW.

Example 3. We do experiments on six UCI databases. The six databases' names and attributes are shown in Table 6. Table 7 is the running time of the feature-weight learning algorithm for the six UCI databases. Table 8 is a WFCM clustering result for Boston database. Table 9 is the clustering results by FCM and WFCM algorithm for the six databases. We can conclude that:

(1) Not all validity functions in Table 9 are simultaneously best. The Boston database's clustering result, for example, is shown in Table 8. When the number of clusters is 2, values of three validity functions are best and the value of V_{fs} is not best. We tend to choose this situation as the best, which has the most optimized indexes.

(2) The amount of improvement for these validity functions is dependent on the specified database and the specified features. For example the improvement of MPG database is quite significant.

Table 6
The characters of some UCI databases

Database name	Number of samples	Number of features	Category of features
Bupa	345	6	Numerical
Boston	506	12	Numerical
Iris	150	4	Numerical
MPG	392	7	Numerical
Pima	768	8	Numerical
Thyroid	215	5	Numerical

Table 7
The feature-weight learning time of the six UCI databases (s)

Database name	Boston	Bupa	Iris	MPG	Pima	Thyroid
Time	521.7	132.1	5	200	640	33

Table 8
The WFCM clustering results of Boston database

Number of clusters	The value of validity functions			
	V_{pc}	V_{pe}	V_{xb}	V_{fs}
2	0.90	0.27	0.06	48,365,212
3	0.81	0.52	0.27	24,788,088
4	0.83	0.48	0.11	−3,546,293
5	0.78	0.64	0.23	−7,369,325
6	0.77	0.70	0.18	−8,859,375

Table 9
Compare the clustering results of the six UCI databases by FCM and WFCM algorithm

		V_{pc}	V_{pe}	V_{xb}	V_{fs}	Number of clusters
Boston	FCM	0.88	0.30	0.07	55,390,708	2
	WFCM	0.90	0.27	0.06	48,365,212	2
Bupa	FCM	0.83	0.42	0.13	81,700.12	2
	WFCM	0.84	0.40	0.12	65,016.43	2
Iris	FCM	0.78	0.57	0.13	−442.94	3
	WFCM	0.86	0.36	0.05	−529.11	3
MPG	FCM	0.74	0.69	0.14	−35	3
	WFCM	0.86	0.41	0.04	−48	3
Pima	FCM	0.82	0.43	0.12	−82,6046	2
	WFCM	0.87	0.31	0.08	−2,297,120	2
Thyroid	FCM	0.64	0.89	0.26	−9490	4
	WFCM	0.69	0.80	0.23	−2190	3

(3) Time complexity is the main problem. WFCM algorithm which improves the performance of FCM is at the price of learning feature-weight. Table 7 shows the feature-weight learning time for the six selected UCI databases. The time complexity of WFCM is $O(cn^2)$ where n is the number of samples and c is a constant associated with the number of features. The learning algorithm can be divided into two parts. One part is searching an appropriate value for β and η which is completed by the one-dimensional searching technique, for example Fiboonacci algorithm. The time complexity of this part is $O(n)$. The other part is ((3.4)–(3.6)) and its time complexity is $O(cn^2)$. In this part, the time complexity depends on the convergence of iterations which are the traditional gradient-descent technique. It is well-known that the gradient-desent algorithm is convergent if the steps are appropriately small. But too small steps will make the convergence rate very slow. Therefore we use Fiboonacci one-dimensional search technique to speed up the convergence rate. Table 7 experimentally shows that the convergence rate for six selected databases. Overall, the time complexity of WFCM is $O(cn^2)$ and its convergence rate is acceptable when the database is not quite big.

From the above three examples we could see that WFCM algorithm indeed improves the performance of FCM.

5. Conclusions

FCM is one of the most well-known clustering algorithms. But its performance has been limited by Euclidean distance. In this paper we propose Weighted FCM algorithm which is based on weighted Euclidean distance. The weighted Euclidean distance incorporates feature-weights into the commonly used Euclidean distance. It shows that an appropriate assignment for feature-weights can improve the performance of FCM clustering. Experiments on some UCI databases illustrate the improvements. We have the following remarks:

(1) Feature-weight learning is an extension of feature selection. Intuitively feature-weight learning techniques are expected to have a better performance than feature selection techniques.
(2) The proposed WFCM can improve the performance of FCM. The improvement is at the price of feature-weight learning which has $O(cn^2)$ time complexity.
(3) The amount of improvement of WFCM over FCM depends on the specific structures of databases.
(4) The investigation to the proposed WFCM can be extended to a sensitivity study of performance of FCM to the selection of distance metric.
(5) A number of extensions of FCM, such as G–K and AFC which use the Mahalanobis distance metric, have been given in literatures. The comparative study on these FCMs based on different metric is in progress.

Acknowledgements

This research is supported by the key project of Chinese Educational Ministry (03017), the Natural Science Foundation of HeBei Province (603137), and the Doctoral Foundation of Hebei Educational Committee (B2003117).

References

Basak, J., De, R.K., Pal, S.K., 1998. Unsupervised feature selection using a neuro-fuzzy approach. Pattern Recog. Lett. 19, 997–1006.

Basu, S., Micchelli, C.A., Olsen, P., 2000. Maximum entropy and maximum likelihood criteria for feature selection from multivariate data. Proc. IEEE Int. Symp. Circuits Syst. III, 267–270.

Bezdek, J.C., 1974. Cluster validity with fuzzy sets. J. Cybernetics 3 (3), 58–73.

Bezdek, J.C., 1975. Mathematical models for systematic and taxonomy. In: Proceedings of 8th International Conference on Numerical Taxonomy, San Francisco, pp. 143–166.

Bezdek, J.C., 1981. Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum, New York.

Bezdek, J.C., Windham, M., Ehrlich, R., 1980. Statistical parameters of fuzzy cluster validity functionals. Comput. Inform. Sci. 9, 232–236.

Cannon, R.L., Dave, J., Bezdek, J.C., 1986. Efficient implementation of the fuzzy *c* means clustering algorithms. IEEE Trans. Pattern Anal. Machine Intell. 8, 248–255.

Dash, M., Liu, H., 2000. Unsupervised feature selection. In: Proceedings of Pacific Asia Conference on Knowledge Discovery and Data mining, pp. 110–121.

De Luca, A., Termini, S., 1972. A definition of a nonprobabilistic entropy in the setting of fuzzy set theory. Inform. Control 20, 301–312.

Dubes, R.C., Jain, A.K., 1988. Algorithms for Clustering Data. Precntice Hall, Englewood Cliffs, NJ.

Dunn, J.C., 1974. Some recent investigations of a new fuzzy partition algorithm and its application to pattern classification problems. J. Cybernetics 4, 1–15.

Dunn, J.C., 1976. Indices of partition fuzziness and the detection of clusters in large data sets. In: Gupta, M.M. (Ed.), Fuzzy Automata and Decision Process. Elsevier, New York.

Dy, J., Brodely, C., 2000. Feature subset selection and order identification for unsupervised learning. In: Proceedings of 17th International Conference on Machine Learning.

Fisher, R., 1936. The use of multiple measurements in taxonomic problems. Ann. Eugenics 7, 179–188.

Fukuyama, Y., Sugeno, M., 1989. A new method of choosing the number of clusters for the fuzzy *c*-means method. In: Proceedings of 5th Fuzzy System Symposium, pp. 247–250.

Gustafson, D.E., Kessel, W., 1979. Fuzzy clustering with a fuzzy covariance matrix. In: Proceedings of IEEE Conference on Decision Control, San Diego, CA, pp. 761–766.

Hall, L.O., Bensaid, A.M., Clarke, L.P., et al., 1992. A comparison of neural network and fuzzy clustering techniques in segmentation magnetic resonance images of the brain. IEEE Trans. Neural Networks 3, 672–682.

Krishnapuram, R., Kim, J., 1999. A note on the Gustafson–Kessel and adaptive fuzzy clustering algorithms. IEEE Trans. Fuzzy Syst. 7, 453–461.

Pal, S.K., Wang, P.P., 1996. Genetic Algorithms for Pattern Recognition. CRC Press, Boca Raton.

Pal, S.K., De, R.K., Basak, J., 2000. Unsupervised feature evaluation: A neuro-fuzzy approach. IEEE Trans. Neural Networks 11, 366–376.

UCI repository of machine learning databases and domain theories. FTP address: www.ics.uci.edu/~mlearn.

Wu, K.L., Yang, M.S., 2002. Alternative *c*-means clustering algorithms. Pattern Recog. 35, 2267–2278.

Xie, X.L., Beni, G.A., 1991. Validity measure for fuzzy clustering. IEEE Trans. Pattern Anal. Machine Intell. 3 (8), 841–846.

Yeung, D.S., Wang, X.Z., 2002. Improving performance of similarity-based clustering by feature weight learning. IEEE Trans. Pattern Anal. Machine Intell. 24 (4), 556–561.

Zhao, S.Y., 1987. Calculus and Clustering. China Renming University Press.