A NEW FEATURE WEIGHTED FUZZY C-MEANS CLUSTERING ALGORITHM

Huaiguo Fu, Ahmed M. Elmisery
Telecommunications Software & Systems Group
Waterford Institute of Technology, Waterford, Ireland

In the field of cluster analysis, most of existing algorithms assume that each feature of the samples plays a uniform contribution for cluster analysis. Considering different features with different importance, feature-weight assignment can be regarded as a special case of feature selection. That is, the feature assigned a value in the interval [0, 1] indicating the importance of that feature, we call this value "feature-weight". In this paper we propose a new feature weighted fuzzy c-means clustering algorithm in a way which this algorithm be able to obtain the importance of each feature, and then use it in appropriate assignment of feature-weight. These weights incorporated into the distance measure to shape clusters based on variability, correlation and weighted features.

Keyword: cluster analysis, fuzzy clustering, feature weighted

1. Introduction

The Goal of cluster analysis is to assign data points with similar properties to the same groups and dissimilar data points to different groups [3]. Generally, there are two main clustering approaches i.e. crisp clustering and fuzzy clustering. In the crisp clustering method the boundary between clusters is clearly defined. However, in many real cases, the boundaries between clusters cannot be clearly defined. Some objects may belong to more than one cluster. In such cases, the fuzzy clustering method provides a better and more useful method to cluster these objects [2]. Cluster analysis has been widely used in a variety of areas such as data mining and pattern recognition [e.g.1, 4, 6]. Fuzzy c-means (*FCM*) proposed by [5] and extended by [4] is one of the most well-known methodologies in clustering analysis. Basically *FCM* clustering is dependent on 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*, so that the obtained clusters are not logically satisfying. Since in most real world problems, features are not considered to be equally important. Considering example in [17], the Iris database [9] 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 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.

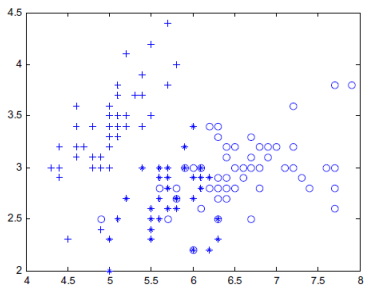

Fig.1. Clustering result of Iris database based on feature weights (1, 1, 0, 0) by *FCM* algorithm.

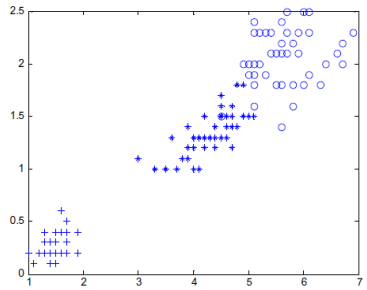

Fig.2. Clustering result of Iris database based on feature weights (0, 0, 1, 1) by *FCM* algorithm.

Feature selection and weighting have been hot research topics in cluster analysis. Desarbo [8] introduced the *SYNCLUS* algorithm for variable weighting in *k*-means clustering. It is divided into two stages. First it uses *k*-means clustering with initial set of weights to partition data into *k* clusters. It then determines a new set of optimal weights by optimizing a weighted mean-square. The two stages iterate until they obtain an optimal set of weights.

Huang [7] presented *W-k*-means, a new *k*-means type algorithm that can calculate variable weights automatically. Based on the current partition in the iterative *k*-means clustering process, the algorithm calculates a new weight for each variable based on the variance of the within cluster distances. The new weights are used in deciding the cluster memberships of objects in the next iteration. The optimal weights are found when the algorithm converges. The weights can be used to identify important variables for clustering. The variables which may contribute noise to the clustering process can be removed from the data in the future analysis.

With respect to *FCM* clustering, it is sensitive to the selection of distance metric. Zhao [12] stated that the Euclidean distance give good results when all clusters are spheroids with same size or when all clusters are well separated. In [13, 10], they 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 [11], 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 that must be incorporated into the Euclidean distance. Each feature should have an importance degree which is called feature-weight. Feature-weight assignment is an extension of feature selection [17]. The latter has only either 0-weight or 1-weight value, while the former can have weight values in the interval [0.1]. Generally speaking, feature selection method cannot be used as feature-weight learning technique, but the inverse is right. To be able to deal with such cases, we propose a new *FCM* Algorithm that takes into account weight of each feature in the data set that will be clustered. After a brief review of the *FCM* in section 2, a number of features ranking methods are described in section 3. These methods will be used in determining *FWA (feature weight assignment)* of each feature. In section 4 distance measures are studied and a new one is proposed to handle the different feature-weights. In section 5 we proposed the new *FCM* for clustering data objects with different feature-weights.

2. Fuzzy C-Mean Algorithm

Fuzzy *c*-mean (*FCM*) is an unsupervised clustering algorithm that has been applied to wide range of problems involving feature analysis, clustering and classifier design. *FCM* has a wide domain of applications such as agricultural engineering, astronomy, chemistry, geology, image analysis, medical diagnosis, shape analysis, and target recognition [14]. Unlabeled data are classified by minimizing an objective function based on a distance measure and clusters prototype. Although the description of the original algorithm dates back to 1974 [4, 5] derivatives have been described with modified definitions for the distance measure and prototypes for the cluster centers [12, 13, 11, 10] as explained above. The *FCM* minimizes an objective function J_m, which is the weighted sum of squared errors within groups and is defined as follows:

$$J_m(U,V;X) = \sum_{k=1}^{n}\sum_{i=1}^{n} u_{ik}^{m} \|x_k - v_i\|_A^2, 1 < m < \infty \quad (1)$$

Where V= $(v_1, v_2, \ldots\ldots, v_c)$ is a vector of unknown cluster prototype (centers) $v_i \in \Re^p$. The value of u_{ik} represent the grade of membership of data point x_k of set X= $\{x_1, x_2, \ldots\ldots, x_c\}$ to the *i*th cluster. The inner product defined by a distance measure matrix *A* defines a measure of similarity between a data object and the cluster prototypes. A hard fuzzy c-means partition of *X* is conveniently represented by a matrix $U = [u_{ik}]$. It has been shown by [4] that if $\|x_k - v_i\|_A^2 > 0$ for all *i* and *k*, then (U,V) may minimize J_m only, when *m>1* and

$$v_i = \frac{\sum_{k=1}^{n} (u_{ik})^m x_k}{\sum_{k=1}^{n} (u_{ik})^m} \quad \text{For } 1 \le i \le c \quad \textbf{(2)}$$

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{\|x_k - v_i\|_A^2}{\|x_k - v_j\|_A^2} \right)^{\frac{1}{m-1}}} \quad \text{For } 1 \le i \le c, 1 \le k \le n \quad \textbf{(3)}$$

Among others, J_m can be minimized by Picard iteration approach. This method minimizes J_m by initializing the matrix ***U*** randomly and computing the cluster prototypes *(Eq.2)* and the membership values *(Eq.3)* after each iteration. The iteration is terminated when it reaches a stable condition. This can be defined for example, when the changes in the cluster centers or the membership values at two successive iteration steps is smaller than a predefined threshold value.

The *FCM* algorithm always converges to a local minimum. A different initial guess of u_{ij} may lead to a different local minimum. Finally, to assign each data point to a specific cluster, defuzzification is necessary, e.g., by attaching a data point to a cluster for which the value of the membership is maximal [14].

3. Estimating *FWA* of features

In section 1 we mentioned that we propose a new clustering algorithm for a data objects with different feature-weights, which means that data with features of different *FWA* should be clustered. A key question that arises here is how we can determine the importance of each feature. In other words, we are about to assign a weight to each feature so that the weight of each feature determines the *FWA* of it.

To determine the *FWA* of features of a data set two major approaches can be adopted: **Human-based approach** and **Automatic approach**. In human-based approach we determine the *FWA* of each feature based on negotiation with an expert individual who has enough experience and knowledge in the field that is the subject of clustering. On the other hand, in automatic approach we use the data set itself to determine the *FWA* of its features. We will discuss more about these approaches in next lines.

Human-based approach: As is described above, in human-based approach by negotiating with an expert, we choose *FWA* of each feature. This approach has some advantages and some drawbacks. In some cases, using the data set itself to determine the *FWA* of each feature may fail to achieve the real *FWA*'s, and human-based approach should be adopted to determine the *FWA* of each feature. Fig.3 demonstrates a situation this case happens.

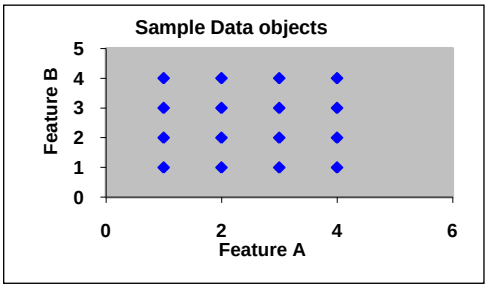

Fig.3. data object with two features

Suppose Fig.3 shows a data objects in which *FWA* of feature ***A*** is two times *FWA* of feature ***B*** in reality. Since automatic approach uses the position of data points in the data space to determine the *FWA* of features, using data set itself to determine the *FWA* of features ***A*** and ***B*** (automatic approach) will lead to equal *FWA*'s for ***A*** and ***B***. Although this case (data set with homogeneously and equidistantly distributed data points) rarely happens in real world and is somehow an exaggerated one, it shows that, sometimes, human-base approach is the better choice.

On the other hand, human-based approach has its own drawbacks. We cannot guarantee that the behaviors that are observed by a human expert and used to determine the *FWA's* include all situations that can occur

due to disturbances, noise, or plant parameter variations. Also suppose situation in which there is no human expert for negotiation to determine *FWA's.* How does this problem should be dealt with?
Structure the signal can be found using linear transforms. This approach does not take into account that the system has some structure. In the time domain, filtering is a linear transformation. The Fourier, Wavelet, and Karhunen-Loeve transforms have compression Capability and can be used to identify some structure in the signals. When we are using these transforms, we do not take into account any structure in the system.

Automatic approach: Several methods based on fuzzy set theory, artificial neural network, fuzzy-rough set theory, principle component analysis and neuro-fuzzy methods and have been reported [16] for weighted feature estimation. Some of the mentioned methods just rank features, but with some modifications they will be able to calculate the *FWA* of the features. Here we introduce a feature weight estimation method which can be used to determine the *FWA* of features. This method extends the one proposed in [15].
Let the *p*th pattern vector (each pattern is a single data item in the data set and a pattern vector is a vector which its elements are the values that the pattern features assume in the data set) be represented as

$$x^p = [x_1^p, x_2^p, \ldots\ldots\ldots, x_n^p] \qquad \textbf{(4)}$$

Where *n* is the number of features of the data set, and x_i^p is the *i*th element of the vector. Let $prob_k$ and $d_k(x^p)$ stand for the priori probability for the class C_k and the distance of the pattern x^p from the *k*th mean vector,

$$m_k = [m_{k1}, m_{k2}, \ldots\ldots\ldots, m_{kn}] \qquad \textbf{(5)}$$ respectively.

The feature estimation index for a subset (Ω) containing few of these *n* features is defined as

$$E = \sum_{x^p \in C_k} \sum_{k} \frac{s_k(x^p)}{\sum_{k' \neq k} s_{k'k}(x^p)} \times \alpha_k \qquad \textbf{(6)}$$

Where x^p is constituted by the features of Ω only.

$$s_k(x^p) = \mu_{ck}(x^p) \times \left(1 - \mu_{ck}(x^p)\right) \qquad \textbf{(7)}$$ and

$$s_{k'k}(x^p) = \frac{1}{2} \times \left[\mu_{ck}(x^p) \times \left(1 - \mu_{ck'}(x^p)\right)\right] + \frac{1}{2} \times \left[\mu_{ck'}(x^p) \times \left(1 - \mu_{ck}(x^p)\right)\right] \qquad \textbf{(8)}$$

$\mu_{ck}(x^p)$ and $\mu_{ck'}(x^p)$ are the membership values of the pattern x^p in classes C_k and $C_{k'}$, respectively. α_k is the normalizing constant for class C_k which takes care of the effect of relative sizes of the classes. Note that s_k is zero (minimum) if $\mu_{ck} = 1$ or 0, and is 0.25 (maximum) if $\mu_{ck} = 0.5$. On the other hand, $s_{k'k}$ is zero (minimum) when $\mu_{ck} = \mu_{ck'} = 1$ or 0, and is 0.5 (maximum) for $\mu_{ck} = 1$, $\mu_{ck'} = 0$ or vice versa.
Therefore, the term $s_k \Big/ \sum_{k \neq k'} s_{k'k}$, is minimum if $\mu_{ck} = 1$ and $\mu_{ck'} = 0$ for all $k \neq k'$ i.e., if the ambiguity in the belongingness of a pattern x^p to classes C_k and $C_{k'}$ is minimum (pattern belongs to only one class). It takes its maximum value when $\mu_{ck} = 0.5$ for all *k*. In other words, the value of **E** decreases as the belongingness of the patterns increases to only one class (i.e., compactness of individual classes increases) and at the same time decreases for other classes (i.e., separation between classes increases). **E** increases when the patterns tend to lie at the boundaries between classes (i.e. $\mu \longrightarrow 0.5$). The objective in feature selection problem, therefore, is to select those features for which the value of ***E*** is minimum [15]. In order to achieve this, the membership $\mu_{ck}(x^p)$ of a pattern x^p to a class is defined, with a multi-dimensional π - function which is given by

$$\mu_{ck}(x^p) \begin{cases} = 1 - 2d_k^2(x^p) & if\ 0 \le d_k^2(x^p) < 0.5 \\ = 2[1 - d_k(x^p)]^2 & if\ 0.5 \le d_k^2(x^p) < 1 \\ = 0 & otherwise \end{cases} \quad \textbf{(9)}$$

The distance $d_k(x^p)$ of the pattern x^p from m_k (the center of class C_k) is defined as:

$$d_k(x^p) = \left[\sum_i \left(\frac{x_i^p - m_{ki}}{\lambda_{ki}} \right)^2 \right]^{1/2}, \quad \textbf{(10)} \qquad \text{where}$$

$$\lambda_{k_i} = 2 \max_p \left(\left| x_i^p - m_{ki} \right| \right) \quad \textbf{(11)}$$

And $m_{ki} = \dfrac{\sum_{p \in C_k} x_i^p}{|C_k|}$ **(12)**

Let us now explain the role of α_k . **E** is computed over all the samples in the feature space irrespective of the size of the classes. Therefore, it is expected that the contribution of a class of bigger size (i.e. with larger number of samples) will be more in the computation of **E**. As a result, the index value will be more biased by the bigger classes; which might affect the process of feature estimation. In order to overcome this i.e., to normalize this effect of the size of the classes, a factor α_k , corresponding to the class C_k , is introduced. In the present investigation, we have chosen $\alpha_k = 1/|C_k|$. However, other expressions like $\alpha_k = 1/prob_k$ or $\alpha_k = 1 - prob_k$ could also have been used.

If a particular subset (F_1) of features is more important than another subset (F_2) in characterizing / discriminating the classes / between classes then the value of ***E*** computed over F_1 will be less than that computed over F_2. In that case, both individual class compactness and between class separation would be more in the feature space constituted by F_1 than that of F_2. In the case of individual feature ranking (that fits to our need for feature estimation), the subset F contains only one feature [15].

Now, using feature estimation index we are able to calculate the *FWA* of each feature. As mentioned above, the smaller the value of ***E*** of a feature, the more significant that feature is. On the other hand, with *FWA* we mean that the larger its value for a given feature, the more significant that feature is. So we calculate the *FWA* of a feature this way: suppose $a_1, a_2, \ldots\ldots a_n$ are *n* features of a data set and *E* (a_i) and *FWA* (a_i) are feature estimation index and feature-weight assignment of feature a_i, respectively so

$$FWA(a_i) = \frac{\left(\sum_{j=1}^{n} E(a_j) \right) - E(a_i)}{\sum_{j=1}^{n} E(a_j)}, \qquad 1 \le i \le n \qquad \textbf{(13)}$$

With this definition, *FWA* (a_i) is always in the interval [0.1]. So we define vector *FWA* which its *i*th element is *FWA* (a_i). Till now we have calculated *FWA* of each feature of the data set. Now we should take into account these values in calculating the distance between data points, which is of great significance in clustering.

4. Modified Distance Measure for the New *FCM* Algorithm

Two distance measures are used in *FCM* widely in literature: Euclidian and Mahalanobis distance measure. Suppose *x* and *y* are two pattern vectors (we have introduced pattern vector in section 3). The Euclidian distance between *x* and *y* is:

$$d^2(x, y) = (x - y)^T (x - y) \quad \textbf{(14)}$$

And the Mahalanobis distance between *x* and a center *t* (taking into account the variability and correlation of the data) is:

$$d^2(x,t,C)=(x-t)^T C^{-1}(x-t) \quad \textbf{(15)}$$

In Mahalanobis distance measure *C* is the co-variance matrix. Using co-variance matrix in Mahalanobis distance measure takes into account the variability and correlation of the data. To take into account the weight of the features in calculation of distance between two data points we suggest the use of $(x\text{-}y)_m$ (modified *(x-y)*) instead of *(x-y)* in distance measure, whether it is Euclidian or Mahalanobis. $(x\text{-}y)_m$ is a vector that its *i*th element is obtained by multiplication of *i*th element of vector (*x – y)* and *i*th element of vector ***FWA***. So, with this modification, *equ.14* and *equ.15* will be modified to this form:

$$d_m^2(x,y)=(x-y)_m^t(x-y)_m \quad \textbf{(16)}$$ and

$$d_m^2(x,t,C)=(x-t)_m^t\ C^{-1}(x-t)_m \quad \textbf{(17)}$$ respectively , where

$$(x-y)_m(i)=(x-y)(i)\times FFWI(i) \quad \textbf{(18)}.$$

We will use this modified distance measure in our algorithm of clustering data set with different feature-weights in next section. To illustrate different aspects of the distance measures mentioned above let's look at some graphs in Fig.4 Points in all graphs are at equal distance (with different distance measures) to the center. A circumference in graph **A** represents points with equal Euclidian distance to the center. In graph **B**, points are of equal Mahalanobis distance to the center. Here the co-variance matrix is: $C=\begin{pmatrix}1 & 0\\ 0 & 4\end{pmatrix}$ In this case the variable **Y** has more variability than the variable **X**, then, even if the values in the y-axis appear further from the origin with respect to the Euclidean Distance, they have the same Mahalanobis distance as those in the x-axis or the rest of the ellipsoid.

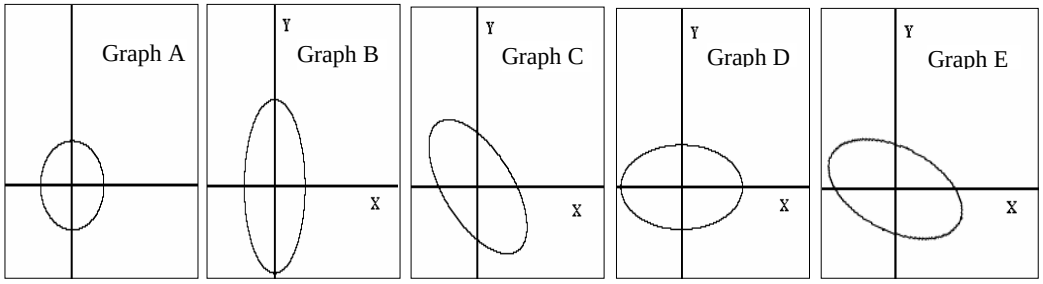

Fig.4. Point with equal distance to the center

In the third case, let's assume that the parameters C is given by $C=\begin{pmatrix}2.5 & -1.5\\ -1.5 & 2.5\end{pmatrix}$ Now the variables have a covariance different from zero. As a consequence, the ellipsoid rotates and the direction of the axis is given by the eigenvectors of **C**. In this case, greater values of **Y** are associated with smaller values of **X**. In other words, every time we move up, we also move to the left, so the axis given by the *y*-axis rotates to the left (see graph (**C**)). Graphs **D** and **E** demonstrate point with equal modified Euclidian and modified Mahalanobis distance to the centre, respectively. In both of them ***FWA*** vector is **FWA=** (0.33 0.67), and in graph **E**, **C** is equal to what it was in graph **C**. Comparing graphs **C** and **E**, we can conclude that in graph **E** in addition to variability and correlation of data, the *FWA* of features is considered in calculating distances.

5. New Feature Weighted *FCM* Algorithm

In this section we propose the new clustering algorithm, which is based on *FCM* and extend the method that is proposed by [15] for determining *FWA* of features and, moreover, uses modified Mahalanobis measure of distance, which takes into account the *FWA* of features in addition to variability of data. As mentioned before, despite *FCM*, this algorithm clusters the data set based on weights of features. In the first step of this algorithm we should calculate the ***FWA*** vector using method proposed in [15]. To do so, we need some clusters over the data set to be able to calculate m_{k_i} and $d_k(x^p)$ (having these parameters in hand, we can easily calculate the feature estimation index for each feature. see section 3). To have these clusters we apply *FCM* algorithm with Euclidian distance on the data set. The created clusters help us to calculate the ***FWA***

vector. This step, in fact, is a pre-computing step. In the next and final step, we apply our Feature weighted *FCM* algorithm on the data set, but here we use modified Mahalanobis distance in *FCM* algorithm.
The result will be clusters which have two major difference with the clusters obtained in the first step. The first difference is that the Mahalanobis distance is used. It means that the variability and correlation of data is taken into account in calculating the clusters. The second difference, that is the main contribution of this investigation, is that features weight index has a great role in shaping the clusters.

6. Conclusions

In this paper, we have presented a new clustering algorithm based on fuzzy c-mean algorithm which is salient feature is that it clusters data set based on weighted features. We used a feature estimation index to obtain *FWA* of each feature. The index is defined based on the aggregated measure of compactness of the individual classes and the separation between the classes in terms of class membership functions. The index value decreases with the increase in both the compactness of individual classes and the separation between the classes. To calculate the feature estimation index we passed a pre-computing step which was a fuzzy clustering using *FCM* with Euclidian distance. Then we transformed the values into the *FWA* vector which its elements are in interval [0, 1] and each element shows the relative significance of its peer feature. Then, we merged the *FWA* vector and distance measures and used this modified distance measure in our algorithm. The result was a clustering on the data set in which weight of each feature plays a significant role in forming the shape of clusters.

References

1. 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.
2. Hung M, D. ang D, 2001 *"An efficient fuzzy c-means clustering algorithm"*. In Proc. the 2001 IEEE International Conference on Data Mining.
3. Han J., Kamber M., 2001 *"Datamining: Concepts and Techniques"*. Morgan Kaufmann Publishers, San Francisco.
4. Bezdek, J.C., 1981. *"Pattern Recognition with Fuzzy Objective Function Algorithms"*. Plenum, New York.
5. Dunn, J.C., 1974. *"Some recent investigations of a new fuzzy partition algorithm and its application to pattern classification problems"*. J. Cybernetics
6. Cannon, R.L., Dave, J., Bezdek, J.C., 1986. *"Efficient implementation of the fuzzy c means clustering algorithms"*. IEEE Trans. Pattern Anal. Machine Intell
7. Huang JZ , Ng MK , Rong H and Li Z.,2005. *"Automated Variable Weighting in k-Means Type Clustering"*. IEEE Transactions on Pattern Analysis & Machine Intelligence, Vol. 27, No. 5.
8. Desarbo W.S., Carroll J.D.; Clark, and Green P.E., 1984 *"Synthesized Clustering: A Method for Amalgamating Clustering Bases with Differential Weighting Variables,"* Psychometrika, vol. 49.
9. Fisher, R., 1936. *"The use of multiple measurements in taxonomic problems"*. Ann. Eugenics 7.
10. Krishnapuram, R., Kim, J., 1999. *"A note on the Gustafson–Kessel and adaptive fuzzy clustering algorithms"*. IEEE Trans. Fuzzy Syst. 7.
11. Wu, K.L., Yang, M.S., 2002. *"Alternative c-means clustering algorithms"*. Pattern Recog. 35.
12. Zhao, S.Y., 1987. *"Calculus and Clustering"*. China Renming University Press.
13. Gustafson, D.E., Kessel, W., 1979. *"Fuzzy clustering with a fuzzy covariance matrix"*. In: Proceedings of IEEE Conference on Decision Control, San Diego, CA.
14. Hopner , K, R., Runkler, 1999 *"Fuzzy Cluster Analysis"*, John Wily & sons.
15. Pal S. K. and Pal A. (Eds.) 2002, *"Pattern Recognition: From Classical to Modern Approaches"*. World Scientific, Singapore.
16. de Oliveira J.V., Pedrycz W., 2007, *"Advances in Fuzzy Clustering and its Applications"*, John Wily & sons.
17. X. Wang, Y. Wang and L. Wang.,2004 "*Improving fuzzy c-means clustering based on feature-weight learning*"**,** Pattern Recognition Letters **25**.