A Fast Fuzzy Clustering Algorithm

MOH'D BELAL AL- ZOUBI, AMJAD HUDAIB, BASHAR AL-SHBOUL
Department of Computer Information Systems
University of Jordan
Amman
JORDAN

Abstract: - Clustering algorithms have been utilized in a wide variety of application areas. One of these algorithms is the Fuzzy C-Means algorithm (FCM). One of the problems with these algorithms is the time needed to converge. In this paper, a Fast Fuzzy C-Means algorithm (FFCM) is proposed based on experimentations, for improving fuzzy clustering. The algorithm is based on decreasing the number of distance calculations by checking the membership value for each point and eliminating those points with a membership value smaller than a threshold value. We applied FFCM on several data sets. The experiments demonstrate the efficiency of the proposed algorithm.

Key-words: - Clustering, Fuzzy C-Means, Pattern Recognition, Data Mining

1. Introduction[1]

Clustering involves dividing data points into homogeneous classes or clusters so that points in the same cluster are similar as possible, and points in different clusters are as dissimilar as possible. In non-fuzzy or hard clustering, data is divided into crisp clusters, where each data point belongs to exactly one cluster. In practice, however, there are many situations in which the data points could be classified as belonging to one cluster almost as well as to another. Such a situation cannot be catered by hard clustering. Therefore, the separation of the clusters becomes a fuzzy notion, and the representation of real data structures can then be more accurately handled by fuzzy clustering algorithms. Hence, it is necessary to describe the data structure in terms of fuzzy clusters. In fuzzy clustering, the data point can belong to more than one cluster, and membership values which indicate the degree to which the data point belongs for the different clusters are associated with the points [1, 2]. The use of membership values provides more flexibility and makes the clustering results more useful in practice.

The Fuzzy C-Means algorithm (FCM), as one of the best known and the most widely used fuzzy clustering algorithms, has been utilized in a wide variety of applications, such as medical imaging [1, 2], remote sensing [3, 4], data mining [5, 6] and pattern recognition [7, 8]. Its advantages include a straightforward implementation, fairly robust behavior, applicability to multidimensional data, and the ability to model uncertainty within the data. A major disadvantage of its requirements is its need for a large amount of time to converge.

In this paper, we propose a Fast Fuzzy C-Means algorithm (FFCM). The FFCM algorithm features several improvements over the FCM. One of its important features is decreasing the number of calculations by checking the membership value for each point and eliminating these points with membership values smaller than a threshold value. The choice of the appropriate threshold is based on experimentations.

The rest of this paper is organized as follows: Section 2 expresses the FCM algorithm. Section 3 proposes the FFCM, and section 4 demonstrates the experimental works and discusses the results. Conclusions are expressed in Section 5.

2. Fuzzy C-Means Algorithm

The Fuzzy C-means Clustering (FCM) algorithm is a data clustering algorithm in which each data point belongs to a cluster to a degree specified by a membership grade [9, 10, 11].

FCM partitions a collection of n data points x_i, $i = 1, \ldots, n$ into c fuzzy groups, and finds a cluster center in each group such that a cost function of dissimilarity measure is minimized. The major

This research is supported by the deanship of Scientific Research, University of Jordan, Amman – Jordan, project No. 958, 2005/2006.

difference between FCM and hard clustering is that FCM employs fuzzy partitioning such that a given data point can belong to several groups with the degree of belongingness specified by the membership grades between 0 and 1. The membership matrix U is allowed to have elements with values between 0 and 1. However, imposing normalization stipulates that the summation of degrees of belongingness for a data set always be equal to unity:

$$\sum_{i=1}^{c} u_{ij} = 1, \forall j = 1,...,n \quad (1)$$

The cost function (or objective function) for FCM is:

$$J(U, c_1,...,c_c) = \sum_{i=1}^{c} J_i = \sum_{i=j}^{c} \sum_{j}^{n} u_{ij}^m d_{ij}^2, \quad (2)$$

where u_{ij} is between 0 and 1; c_i is the cluster center of fuzzy group i; $d_{ij} = \|c_i - x_j\|$ is the Euclidean distance between ith cluster and jth data point; and $m \in [1, \infty)$ is weighting exponent.

The necessary conditions for Equation (2) to reach its minimum are:

$$c_i = \frac{\sum_{j=1}^{n} u_{ij}^m x_j}{\sum_{j=1}^{n} u_{ij}^m}, \quad (3)$$

and

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}} \right)^{2/(m-1)}} \quad (4)$$

The fuzzy C-means algorithm is simply an iterated procedure through the preceding two necessary conditions. The FCM algorithm determines the cluster centers c_i and the membership matrix U using the following steps [12, 13, 14]:

Step [1]: Initialize the membership matrix U with random values between 0 and 1 such that the constraints in Equation (1) are satisfied.

Step [2]: Calculate c fuzzy cluster centers c_i, $i = 1, \ldots, c$, using Equation (3).

Step [3]: Compute the objective function according to Equation (2). Stop if either it is below a certain tolerance value or its improvement over previous iteration is below a certain threshold, ε.

Step [4]: Compute a new U using Equation (4). Go to step 2.

The most frequent complaint about FCM is that it may consume significant amounts of CPU time to converge [15], when large data sets are concerned.

In [15], an algorithm called AFCM was implemented to speed up the FCM by using a lookup table approach. However, the AFCM is not guaranteed to converge, and the lookup tables depend on the number of bits in the data.

Delaunay triangular functions are used to store proximity information to speed up the clustering process in [16].

In [17], the authors utilize visualization in conjunction with automated clustering to speed up the process of partitioning data.

Several efficient and scalable parallel algorithms have been proposed for a special purpose architecture where a variable number of processors is available [18].

In [19], a description of a modified FCM algorithm known as 2rFCM is given. The algorithm reduces the number of points to be clustered by reducing the precision (quantization) of the data. However, this quantization is often a reduction in precision, and therefore, the loss of information is possible.

3. Fast Fuzzy C-Means Algorithm

This research aims at decreasing the number of distance calculations of the FCM by computing the distances between data points and the nearest cluster centres for points with membership values greater than a threshold, T, where the value of T is less than 1 and greater than 0.

In this case, there is no need to calculate distances for points with membership values less than T since these values do not severely affect the results and therefore, some distance calculations can be saved.

To illustrate the FFCM algorithm, we consider the data set "*fcmdata.dat"* of the MATLAB ® package that consists of 140 points in R^2.

Assume that we want to determine a fuzzy partition with two clusters (i.e., C = 2). Assume also that we choose $T = 0.5$, then we obtain part of the U matrix shown in Table 1. Here, we don't compute distances between the cluster centers and the points for U values less than T (the shaded values in Table 1). For example, the distance between cluster $C2$ and point X(1) in not computed, and hence, some time savings can occur.

Table 1 Fuzzy partition, when $T = 0.5$ and $C = 2$

	X(1)	X(2)	X(3)	…	X(139)	X(140)
C1	0.99	0.01	0.10	…	0.06	0.13
C2	0.01	0.99	0.90	…	0.94	0.87

It is expected that more time savings can be obtained for a larger number of clusters. For

example, when C = 3, more distance calculations between the cluster centres and the corresponding data points can be saved as shown in Table 2, where the shaded cells represent these cases.

Table 2 Fuzzy partition, when T = 0.5 and C = 3

	X(1)	X(2)	X(3)	…	X(139)	X(140)
C1	0.01	0.33	0.63	…	0.09	0.25
C2	0.00	0.65	0.31	…	0.89	0.69
C3	0.99	0.02	0.06	…	0.02	0.06

However, for T = 0.2, there are less shaded cells for the same data set and the same number of clusters as shown in Table 3, and hence, less time savings can be obtained when T is decreased. Therefore, we will decide on the value of T when different data sets are used in the next section.

Table 3 Fuzzy partition, when T = 0.2 and C = 3

	X(1)	X(2)	X(3)	…	X(139)	X(140)
C1	0.01	0.33	0.63	…	0.09	0.25
C2	0.00	0.65	0.31	…	0.89	0.69
C3	0.99	0.02	0.06	…	0.02	0.06

4. Experimental Works

In order to test the efficiency of our proposed algorithm, three data sets have been tested. The first set is the *fcmdata* data set mentioned above. The second set, which contains data points in 2 dimensional formats (32768 x 2), represents the well-known Baboon image, and the third set represents the Cameraman image (32768 x 2). The differences in the values of the objective functions between the results of the FFCM and FCM algorithms are referred to as "Error" in the tables below. Both algorithms started with the same initial values which were chosen randomly from the data points of each data set.

One question is raised: What is the value of the "best" threshold, T, to choose?

To answer the question, a series of tests were performed starting with the first data set. When T = 0.9, the results obtained from the proposed FFCM algorithm are different from those obtained from the FCM for the first data set, as shown in Fig.1, where the (two) cluster centers obtained from the FCM algorithm are marked with 'X', and those obtained from our proposed FFCM are represented by 'O'.

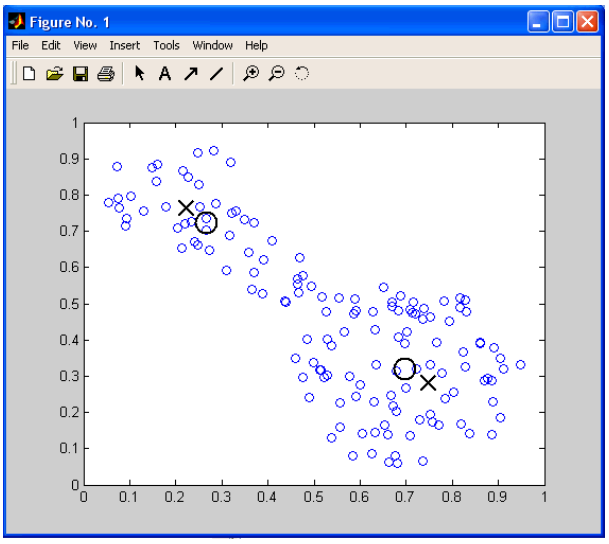

Fig.1 Results obtained from the FFCM and FCM algorithms for T = 0.9

The best results were obtained when T = 0.5, and the results obtained from the FFCM algorithm are the same as the ones obtained from the FCM, as shown in Fig. 2.

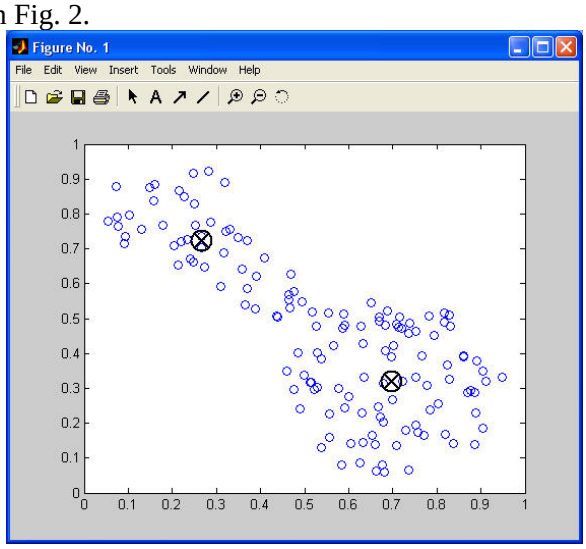

Fig.2 Results obtained from the FFCM and FCM algorithms for T = 0.5

Our experiments showed that the objective function values obtained from our proposed algorithm are the same as those obtained from the FCM when the value of T is equal to 0.5.

Table 4 shows the performance of the proposed (FFCM) algorithm with a different number of clusters and with (T = 0.48) for the second data set. The Table shows that the proposed algorithm gave better time performance than the FCM algorithm. The time savings exceeded 80% in many cases. However, as expected, the FCM algorithm gave better quality results (see the "Error" values in the table).

Table 4: The results of the FFCM with $T = 0.42$

No. of Clusters	Error	Time Savings
10	4.5%	81.7%
20	9.3%	88.9%
30	6.8%	89.0%
40	17.9%	91.1%
50	19.1%	82.2%

Table 5 shows that the results obtained from the proposed algorithm have better time performance in all cases for the third data set, with ($T = 0.28$). However, the quality obtained from our algorithm is degraded.

Table 5: The results of the FFCM with $T = 0.28$

No. of Clusters	Error	Time Savings
10	2.3%	37.1%
20	2.9%	61.7%
30	9.9%	89.6%
40	4.8%	74.0%
50	10.5%	79.2%

Note that the results obtained from Table 5 gave less time performance for the proposed algorithm than those of Table 4. This is because the threshold, *T*, was smaller for the third data set.

5. Conclusions

In this paper, a Fast Fuzzy C-Means (FFCM) algorithm is proposed. One important feature of the FFCM algorithm is decreasing the number of distance calculations required by the FCM algorithm. This is done by checking the membership value for each point and eliminating these points with membership values smaller than a threshold value. The choice of the appropriate threshold value was based on experimentations. The experiments show that the performance of the FFCM algorithm is considered to be better than the performance of the FCM algorithm in terms of the time needed for it to be run. This performance is obtained when a threshold value is within the range (0.28 - 0.5). However, the quality of the results is degraded, specially when the number of clusters decreases.

Acknowledgment

This research is supported by the deanship of Scientific Research, University of Jordan, Amman – Jordan, project No. 958, 2005/2006.

References

[1] D. Pham, Spatial Models for Fuzzy Clustering, *Computer Vision and Image Understanding*, Vol. 84, No. 2, 2001, pp. 285–297.

[2] J. Bezdek, L. Hall, and L. Clarke, Review of MR Image Segmentation Techniques Using Pattern Recognition, *Medical Physics*, Vol. 20, No. 4, 1993, pp. 1033–1048.

[3] E. Rignot, R. Chellappa, and P. Dubois, Unsupervised Segmentation of Polarimetric SAR Data Using the Covariance Matrix, *IEEE Trans. Geosci. Remote Sensing,* Vol. 30, No. 4, 1992, pp. 697–705.

[4] W. Chumsamrong, P. Thitimajshima, and Y. Rangsanseri, Syntetic Aperture Radar (SAR) Image Segmentation Using a New Modified Fuzzy C-Means Algorithm, in *Proceedings of Geoscience and Remote Sensing Symposium*, Vol. 2, 2000, pp. 624–626.

[5] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Second Edition, Morgan-Kaufman, 2006.

[6] M. Dunham, *Data Mining: Introductory and Advanced Topics*, Prentice Hall, 2003.

[7] P. Franti and J. Kivijarvi, Randomised Local Search Algorithm for the Clustering Problem, *Pattern Analysis & Applications,* Vol. 3, 2000, pp.358–369.

[8] B. Ripley, *Pattern Recognition and Neural Networks*, Cambridge University Press, 1996.

[9] K. Wu and M. Yang, Alternative C-Means Clustering Algorithms, *Pattern Recognition,* Vol. 35, No. 10, 2002, pp. 2267-2278.

[10] R. Hathaway and J. Bezdek, Fuzzy C-Means Clustering of Incomplete Data, *IEEE Transactions on Cybernetics,* Vol. 31, No. 5, 2001, pp. 735-744.

[11] L. Hall, A. Bensaid, L. Clarke, R. Velthuizen, M. Silbiger, and J. Bezdek, A Comparison of Neural Network and Fuzzy Clustering Techniques in Segmenting Magnetic Resonance Images of the Brain, *IEEE Transactions on Neural Networks*, Vol. 3, No. 5, 1992, pp. 672-682.

[12] K. Chung, H. Tzeng, S. Chen, J. Wu and T. Chen, Fuzzy C-means Clustering with Spatial Information for Image Segmentation, *Computerized Medical Imaging and Graphics,* Vol. 30, No. 1, 2006, pp. 9-15.

[13] Z. Chi, H. Yan and T. Pham, *Fuzzy Algorithms: With Applications to Image Processing and Pattern Recognition*, World Scientific, 1998.

[14] J. Jang, C. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing*, Prentice-Hall, 1997.

[15] R. Cannon, J. Dave, and J. Bezdek, Efficient Implementation of the Fuzzy C-Means Clustering Algorithms, IEEE Trans. PAMI, Vol. 8, 1986, pp. 248–255.

[16] V. Estivill-Castro and M.Houle, Robust Distance-Based Clustering with Applications to Spatial Data Mining, Algorithmica, Vol. 30, No. 2, 2001, pp. 216–242.

[17] W. Ribarsky, J. Katz, F. Jiang, and A. Holland, Discovery Visualization Using Fast Clustering, IEEE Computer Graphics and Applications, Vol. 19 No. 5, 1999, pp. 32–39.

[18] C. Wu, S. Horng, Y. Chen, and W. Lee, Designing Scalable and Efficient Parallel Clustering Algorithms on Arrays with Reconfigurable Optical Buses. Image and Vision Computing, Vol. 18, No. 13, 2000, pp. 1033–1043.

[19] S. Eschrich, J. Ke, L.. Hall, and D. Goldgof, Fast Fuzzy Clustering of Infrared Images, In B. Gruver, M. Smith, and L. Hall, editors, Joint 9th IFSA World Congress and 20th NAFIPS International Conference, 2001.