On-line Evolving Fuzzy Clustering

V. Ravi[#*], E. R. Srinivas[§] and N. K. Kasabov[$]
[#]*Institute for Development and Research in Banking Technology, Castle Hills Road #1, Masab Tank, Hyderabad – 500 057 (AP) India*
rav_padma@yahoo.com
[§] *65 Hume Avenue, #06-02 Singapore 598743*
erukuravi@yahoo.com
[$] *KEDRI, Auckland University of Technology, Auckland, New Zealand,*
nkasabov@aut.ac.nz

Abstract

In this paper, a novel on-line evolving fuzzy clustering method that extends the evolving clustering method (ECM) of Kasabov and Song (2002) is presented, called EFCM. Since it is an on-line algorithm, the fuzzy membership matrix of the data is updated whenever the existing cluster expands, or a new cluster is formed. EFCM does not need the numbers of the clusters to be pre-defined. The algorithm is tested on several benchmark data sets, such as Iris, Wine, Glass, E-Coli, Yeast and Italian Olive oils. EFCM results in the least objective function value compared to the ECM and Fuzzy C-Means. It is significantly faster (by several orders of magnitude) than any of the off-line batch-mode clustering algorithms. A methodology is also proposed for using the Xie-Beni cluster validity measure to optimize the number of clusters.

1. Introduction

Clustering has been one of the most important tasks in the statistical learning theory [1-3]. Clustering is concerned with finding the grouping of data vectors based on their similarity through defining the cluster centers and their radii. The exact clustering methods (such as K-means) define the membership of each vector as belonging to only one cluster (membership degree of 1) and not belonging to the rest of the clusters (membership degree of 0). In the early 1980's fuzzy C-means algorithm was proposed by Bezdek [2] as an extension of the K-means algorithm to account for the fuzziness present in a data set. Each data vector belongs to every cluster to a certain degree, all degrees having a sum of 1. Several algorithms are proposed thereafter to cater to different shapes of clusters such as Gustafsson and Kessel [4]; Gath and Geva [5]. Clustering algorithms are often used as parts of other methods of computational intelligence, such as data mining and learning algorithms [6], radial basis function neural network to select the kernels [7], fuzzy inference systems where the number of possible fuzzy 'if-then' rules is determined by the number of clusters in the data set [8-10]. Some of the recently reported fuzzy clustering algorithms include collaborative fuzzy clustering [11], robust fuzzy clustering [12], fuzzy clustering based on axiomatic fuzzy set theory [13], threshold accepting based fuzzy clustering [14] and hierarchical approach to fuzzy clustering [15].

When the on-line algorithms became popular over the last decade [16-21], the evolving clustering method ECM emerged as one of them [10]. Kasabov and Song [10] demonstrated the effectiveness of ECM in the context of designing a dynamic, evolving neuro-fuzzy inference system, called DENFIS, which is then applied to modeling and knowledge discovery tasks in bioinformatics, brain study and intelligent machines [22]. Indeed, many applications in the mentioned areas as well as in finance, environmental study, adaptive process control, and other

[*] Corresponding author, Phone: +91-40-2353 4981; FAX: +91-40-2353 5157

0-7695-3050-8/07 $25.00 © 2007 IEEE
DOI 10.1109/ICCIMA.2007.111

areas require a fast on-line clustering to evolve and adapt a model incrementally and continuously to incoming data.

This paper describes a fuzzy version of the ECM, here called "on-line evolving fuzzy clustering method" – EFCM. EFCM results in a much lower objective function value than the ECM or Fuzzy C-Means. It is significantly (several orders of magnitude) faster than the off-line clustering algorithms without the need to specify, in advance, the number of the clusters. The rest of the paper is organized as follows. Section 2 gives a brief description of the evolving clustering method (ECM) of Kasabov and Song [10] and its extension – evolving fuzzy clustering method that incorporates fuzziness in the cluster interpretation stage and a methodology to cluster data sets where the number of clusters is not known *a priori*. Section 3 describes the results of a comparative study of the EFCM versus ECM and Fuzzy-C Means algorithms on 6 different data sets. Section 4 presents conclusions and directions for further research.

2. Evolving Clustering Method (ECM) and The Evolving Fuzzy Clustering Method (EFCM)

2.1. The ECM Algorithm

In this section, first the evolving, online, maximum distance-based clustering method, ECM [10], is described. This was used to implement a scatter partitioning of the input space for creating fuzzy inference rules from data in dynamic evolving neuro-fuzzy inference system (DENFIS). The online ECM does not involve any optimization. It is a fast, one-pass algorithm for a dynamic estimation of the number of clusters and current cluster centers in a data set. In any cluster, the maximum distance, *MaxDis*t, between a sample point and the cluster center, is less than a threshold value, *Dth*r, which is set as a clustering parameter. This parameter affects the number of clusters to be estimated.

In the online clustering process, the data samples come from a data stream and the algorithm starts with an empty set of clusters. When a new cluster is created, the cluster center, Cc, is defined and its cluster radius, Ru, is initially set to zero. When more samples are presented one after another, some existing clusters will be updated by changing their centers' positions and increasing their radii. Which cluster will be updated and how much it will be changed, depends on the position of the current sample in the input space. A cluster will not be updated any more when its cluster radius, Ru, becomes equal to a threshold value, *Dth*r.

2.2. The Proposed Evolving Fuzzy Clustering Method (EFCM)

The evolving fuzzy clustering method (EFCM) is a fuzzy variant of the ECM. It essentially computes the membership values or grades of each of the input vector (sample) as it arrives from the input stream, signifying its degree of belonging to the existing clusters. This variation brings the fuzzy flavor to the ECM and makes it more accurate in case of overlapping clusters and faster vis-à-vis the off-line fuzzy clustering algorithms. The EFCM algorithm is described as follows:

Step 0: Create the first cluster by simply taking the position of the first sample from the input stream as the first cluster center Cc1, and setting a value 0 for its cluster radius Ru1. Create an empty dynamic array to store the membership values of samples in the clusters created as the algorithm proceeds.

Step 1: If all samples have been processed, the algorithm is stopped. Else, the current input sample x_i is considered and the distances between this sample and all n already existing cluster centers Cc_j, $D_{ij} = \|x_i - Cc_j\|, j = 2,...,n$ are calculated. Then, the membership values for this sample is calculated as:

If $x_i \neq Cc_j$ then $\mu_{ij} = \frac{\left(\left\|x_i - Cc_i\right\|\right)^{\frac{-2}{(m-1)}}}{\sum_{j=1}^{n}\left(\left\|x_i - Cc_j\right\|\right)^{\frac{-2}{(m-1)}}} where\ j = 1,2,...,n\ and\ m \in (1,\infty)$

Else $\mu_{ij} = \begin{cases} 1 & for \quad j = i \\ 0 & for \quad j \neq i \end{cases}$ (1)

Step 2: If any distance value, D_{ij}, is equal to or less than, at least one of the radii, Ru_j, $j = 1,2,...,n$, then the current sample x_i belongs to a cluster C_m with the minimum distance

$$D_{im} = \left\|x_m - Cc_m\right\| = min\left\|x_i - Cc_j\right\|, j = 2,...,n \quad (2)$$

subject to the constraint $D_{ij} \leq Ru_j, j = 1,2,...,n$

In this case, neither a new cluster is created, nor any existing cluster is updated. The algorithm returns to Step 1. Else—go to Step 3.

Step 3: Find cluster C_a (with center Cc_a and cluster radius Ru_a) from all n existing cluster centers such that

$$Cc_a = S_{ia} = \min(S_{ij}),\ where\ S_{ij} = D_{ij} + Ru_j, j = 1,2,...,n \quad (3)$$

Step 4: If S_{ia} is greater than *2* x *Dthr*, the sample x_i does not belong to any existing clusters. Hence, a new cluster is created as in Step 0. Then the algorithm returns to Step 2.

Step 5: If S_{ia} is not greater than *2* x *Dthr*, the cluster C_a is updated by moving its center, Cc_a, and increasing the value of its radius, Ru_a. The updated radius is set to be equal to $S_{ia} / 2$ and the new center is located at the point on the line joining x_i and Cc_a, and the distance from the new center to the point x_i is equal to new radius. Further, the membership value matrix is updated to reflect this change. The algorithm returns to Step 1.

End of Algorithm

In the above algorithm, in Step 4, using the formula presented in step 1, the membership value matrix is extended with the addition of a new column to cater to the new cluster. The effectiveness of the ECM and the EFCM is measured by the objective function value that is computed at the end of the clustering process using the following formulas:

(a) In case of ECM:

$$z = \sum_{j=1}^{k}\sum_{i=1}^{n}\left\|x_j - Cc_i\right\|\ , where\ k\ is\ the\ number\ of\ samples \quad (4a)$$

(b) In case of EFCM:

$$z = \sum_{j=1}^{k}\sum_{i=1}^{n}\mu_{ji}\left\|x_j - Cc_i\right\|\ , where\ k\ is\ the\ number\ of\ samples \quad (4b)$$

The less the objective function value the better the clustering is. As it can be seen from the EFCM algorithm, it is much simpler than the traditional fuzzy C-means and much faster than the fuzzy C-means, as it is a one-pass algorithm. Further, it may be observed that the present algorithm utilizes the fuzzy membership value matrix to evaluate the efficacy of the clustering process and does not use it in the clustering process *per se.* This is a significant departure from many fuzzy clustering algorithms. A future version of EFCM would address the issue of incorporating the fuzzy membership value matrix directly into the clustering process.

2.3. A Methodology to Obtain the Optimal Number of Clusters

In datasets with unknown number clusters, the following methodology is suggested for optimizing the number of the clusters in the on-line EFCM:

1. Choose a small value of Dthr, in the range of 0 to 1

2. Run the EFCM algorithm for this value of Dthr
3. Compute the Xie-Beni cluster validity measure (see Xie and Beni [23])
4. Increment the value of Dthr by a small amount and repeat the steps 1 to 3.

The number of clusters corresponding to the minimum value of the Xie-Beni index indicates the "optimal" number of clusters.

3. Results and Discussions

EFCM is used to cluster several benchmark datasets: *Iris, Wine, Glass, E.coli, Yeast* and *Italian Olive Oils* in an on-line mode. The first five data sets are taken from the UCI repository of machine learning databases (Blake and Merz, [24]), while the *Italian Olive Oils* data set is taken from Forina and Armanino [25]. The *Iris* data set has 3 classes of flowers and 150 samples with four features each. The *Wine* data set has 3 classes representing three regions of brewing in Italy. It has 178 samples with 13 features each. The *Glass* data set has 7 classes and 214 samples with 8 features each. The *E.coli* data set has 8 classes with 336 samples and 7 features each. The *Yeast* data set has 10 classes. It has 1484 samples with 8 features each. The *Italian Olive Oils* data set has 9 classes representing the nine regions of growing olive oil in Italy, with 572 samples and 8 features representing the fatty acids. The algorithm is implemented in MATLAB 6.0 [10,26-27].

To compare the performance of the EFCM with the on-line ECM and the off-line Fuzzy-C Means algorithms, these algorithms were also applied on the same date sets. The results presented in Table 1 indicate the consistent superiority of EFCM over both ECM and FCM because the former obtained less objective value in all the data sets. All variables in the data sets were normalized in the interval [0,1] before clustering. This demonstrates the overwhelming superiority of EFCM over both ECM and FCM. Table 1 also presents the number of clusters obtained in each case. According to the methodology presented in section 2.3, Xie-Beni index is used to determine the optimal number of clusters in all cases. Table 1 reveals that both EFCM and ECM reported the same number of clusters for each data set except *Glass* where EFCM yielded fewer clusters compared to both ECM and Fuzzy C-Means. This is due to the fact that the calculated Xie-Beni index value was minimum for 5 clusters in case of EFCM, whereas there were 7 clusters evolved through ECM, even though for the same value of Dthr both EFCM and ECM evolve same number of clusters. This is because the definition of the Xie-Beni index for EFCM involves membership degree values of the samples in each cluster.

Table 1: Comparison of objective function values

Dataset	EFCM (Present paper)		ECM [10]		FCM [2]	
	Objective function	# clusters	Objective function	# clusters	Objective function	# clusters
Iris	**0.0476**	3	0.1529	3	0.0903	3
Wine	**0.3231**	2	0.8887	2	1.2045	3
Glass	**0.1051**	5	0.6862	7	1.9002	7
E.Coli	**0.0930**	8	0.9314	8	0.2060	3
Yeast	**0.0346**	9	0.7140	9	0.8628	3
Italian Olive Oils	**0.1327**	3	0.4926	3	0.6978	3

4. Conclusions

This paper presents a novel on-line fuzzy clustering algorithm called ***evolving fuzzy clustering method*** (EFCM) by extending the evolving clustering method (ECM) [10]. Its effectiveness is tested on several benchmark data sets such as *Iris, Wine, Glass, E.Coli, Yeast* and *Italian Olive*

oils. A methodology to optimize the number of clusters in the EFCM algorithm for clustering the data sets where the number of clusters is unknown *a priori* is also suggested. Results demonstrate the superiority of the present algorithm EFCM over both ECM and FCM as it yielded the least objective function value compared to the latter two algorithms. The EFCM is very fast since it is a one-pass algorithm. Hence it is best suited for on-line data clustering across application areas.

5. References

1. J. C. Bezdek, *Analysis of Fuzzy Information*. 1987: Boca Raton Fla, CRC Press.
2. J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. 1981, New York: Plenum Press.
3. V. Vapnik, Statistical Learning Theory. Adaptive and Learning Systems, ed. S. Haykin. 1998: John Wiley&Sons, Inc. 736.
4. D. Gustafsson and W. Kessel, "Fuzzy Clustering with a Fuzzy Covariance Matrix", *Proceedings of the IEEE CDC*, San Diego, California, IEEE Press, Piscataway, New Jersey, 1979, pp.761-766.
5. I. Gath and A. B. Geva, "Unsupervised optimal fuzzy clustering", *IEEE Trans. on Pattern Analysis & Machine Intelligence*, 1989, Vol. 11, pp. 773-781.
6. V. Cherkassky and F. Mulier, *Learning From Data Concepts, Theory and Methods,* A Volume in The Wiley Series on Adaptive and Learning Systems for Signal Processing, Communications, and Control, ed. S. Haykin. 1997: Wiley-Interscience. 441.
7. J. Moody and C. J. Darken, "Fast learning in networks of locally-tuned processing units", *Neural Computation*, 1989, Vol. 1, pp. 281-294.
8. J. C. Bezdek and S. K. Pal, eds. *Fuzzy Models For Pattern Recognition - Methods That Search for Structures in Data*. IEEE Press, Year 1992, 539.
9. R. R. Yager and L. A. Zadeh, eds. *An Introduction to Fuzzy Logic Applications In Intelligent Systems*, Kluwer Academic Publishers, Year 1992, 356.
10. N. K. Kasabov and Q. Song, "DENFIS: Dynamic, evolving neural-fuzzy inference systems and its application for time-series prediction*", IEEE Trans. on Fuzzy Systems*, 2002. Vol.10, No. 2, pp. 144-154.
11. W. Pedrycz, Collaborative fuzzy clustering, Pattern Recognition Letters, 23, 2002, 1675-1686.
12. J. Leski, Towards robust fuzzy clustering, Fuzzy Sets and Systems, 137, 2003, 215-233.
13. X. Liu, W. Wang and T. Chai, The fuzzy clustering analysis based on AFS theory, IEEE Transaction on Systems, Man and Cybernetics- Part B: Cybernetics, 35, 5, 2005, 1013-1027.
14. V. Ravi, M. Bin and P. Ravi Kumar, Threshold accepting based fuzzy clustering algorithms, International Journal of Uncertainty, Fuzziness and Knowledge-based systems, 14, 5, 2006, 617-632.
15. E. N. Nasibov and G. Ulutagay, A new unsupervised approach for fuzzy clustering, Fuzzy Sets and Systems, 158, 2007, 2118-2133.
16. N. K. Kasabov, On-line learning, reasoning, rule extraction and aggregation in locally optimized evolving fuzzy neural networks*, Neurocomputing*, 2001, Vol. 41, pp. 25-45.
17. G. A. Rummery and M. Niranjan, *On-line Q-learning using connectionist systems*, in Cambridge University Engineering Department. 1994.
18. T. M. Heskes and B. Kappen, On-line learning processes in artificial neural networks in: *Mathematical Foundations of Neural Networks*, Elsevier, Amsterdam, 1993, pp. 199-233.
19. D. Saad, On-line learning in Neural Networks. 1999, Cambridge University Press, Australia.
20. N. Kasabov, Evolving Fuzzy Neural Networks - Algorithms, Applications and Biological Motivation, in: *Methodologies for the conception, design and application of soft computing,* T. Yamakawa and G. Matsumoto, Editors. World Scientific, 1998, pp. 271-274.
21. N. Kasabov, Evolving fuzzy neural networks for on-line supervised/ unsupervised, knowledge-based learning, *IEEE Transactions on Systems, Man and Cybernetics - Part B, Cybernetics*, 2001, Vol. 31, No. 6, pp. 902-918.
22. N. Kasabov, *Evolving Connectionist Systems - Methods and Applications in Bioinformatics, Brain Study and Intelligent Machines*, 2003, Springer-Verlag, London.
23. X. L. Xie and G. Beni, "A validity measure for fuzzy clustering", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 8, 1991, pp. 841-847.
24. C.L.Blake and C.J.Merz, UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science, 1998.
25. M. Forina and C. Armanino, Eigenvector projection and simplified nonlinear mapping of fatty acid content of Italian olive oils, *Annali di Chimica*, Vol. 72, 1982, pp. 127-141. (The olive oil data set is available online from ftp://ftp.clarkson.edu/pub/hopkepk/Chemdata/Original/oliveoil.dat.
26. Math Works, Statistics Toolbox (for use with Matlab). 2002.
27. MathWorks, T., Neural Network Toolbox User's Guide. Vol. 4. 2001: The Math Works Inc.