Improved Fuzzy C-Means Clusters With Ant Colony Optimization

[1]C.Immaculate Mary, [2]Dr. S.V. Kasmir Raja

[1]Associate professor of Computer Science ,Sri Sarada College, Salem-636016, Tamil Nadu, India.

[2]Dean (Research), S.R.M. University , Chennai, Tamil Nadu, India.

E-mail: cimmaculatemary@gmail.com

Abstract: Cluster analysis aims at identifying groups of similar objects and, therefore helps to discover distribution of patterns and interesting correlations in large data sets. These methods are not only major tools to uncover the underlying structures of a given data set, but also promising tools to uncover local input-output relations of a complex system. Fuzzy C-means (FCM) is one of the most widely used fuzzy clustering algorithms in real world applications. However there are two major limitations that exist in this method. The first is that a predefined number of clusters must be given in advance. The second is that the FCM technique can get stuck in sub-optimal solutions. In this paper, we have proposed an ant colony algorithm to improve the clusters obtained from fuzzy c-means clustering. The proposed algorithm is tested in medical domain and the results show that post processing refinement of clusters improves the cluster quality.

Keywords: Fuzzy C-Means, Ant Colony Optimization, Cluster Refinement

1. Introduction

Clustering is one of the most useful tasks in data mining process for discovering groups and identifying interesting distributions and patterns in the underlying data. Clustering problem is about partitioning a given data set into groups (clusters) such that the data points in a cluster are more similar to each other than points in different clusters (Guha, et al., 1998). For example, consider a retail database records containing items purchased by customers. A clustering procedure could group the customers in such a way that customers with similar buying patterns are in the same cluster. Thus, the main concern in the clustering process is to reveal the organization of patterns into "sensible" groups, which allow us to discover similarities and differences, as well as to derive useful conclusions about them. This idea is applicable in many fields, such as life sciences, medical sciences and engineering. Clustering may be found under different names in different contexts, such as unsupervised learning (in pattern recognition), numerical taxonomy (in biology, ecology), typology (in social sciences) and partition (in graph theory) (Theodoridis & Koutroubas, 1999). In the clustering process, there are no predefined classes and no examples that would show what kind of desirable relations should be valid among the data that is why it is perceived as an unsupervised process (Berry & Linoff, 1996). On the other hand, classification is a procedure of assigning a data item to a predefined set of categories (Fayyad, et al., 1996). Clustering produces initial categories in which values of a data set are classified during the classification process.

Clustering analysis is the main component of unsupervised techniques. Recently various algorithms for clustering large data sets and streaming data sets have been proposed (Pal and Bezdek 2002, Ramakrishnan and Livny 1996, Bradley et al., 1998, Farnstrom et al., 2000, Guha et al., 1998, Ng and Han 2002, Gupta and Grossman 2004, O'Callaghan et al., 2002). The focus has been primarily either on sampling (Pal and Bezdek 2002, Guha et al., 1998, Ng and Han 2002, Hathaway and Bezdek, 2006) or incrementally loading partial data, as much as can fit into memory at one time. The incremental approach (Bradley et al., 1998, Farnstrom et al., 2000, Gupta and Grossman 2004, O'Callaghan et al., 2002) generally keeps sufficient statistics or past knowledge of clusters from a previous run of a clustering algorithm in some data structures and uses them in improving the model for the future.

Clustering can also be performed in two different modes: crisp and fuzzy. In crisp clustering, the clusters are disjoint and non-overlapping in nature. Any pattern may belong to one and only one class in this case. In case of fuzzy clustering, a pattern may belong to all the classes with a certain fuzzy membership grade (Jain et al., 1999). A common fuzzy clustering algorithm is the Fuzzy C-Means (FCM), an extension of classical C Means algorithm for fuzzy applications (Bezdeck et al., 1984). The FCM method (Canno et al., 1986, Kamel and Selim, 1994), suffer several difficulties: a) sensitive to the initialization; b) inability to find a global minimum and; c) difficulty of deciding how many clusters exist. Since FCM's performance depends on selected metrics, it will depend on the feature- weights which are incorporated into the Euclidean distance. Wang et al., (2004) try to adjust these feature weights to improve FCM's performance. Wang and Garibaldi (2005) proposed an alternative fuzzy clustering algorithm, Simulated Annealing Fuzzy Clustering (SAFC) that improves the cluster quality. Various algorithms (Cheng et al., 1998, Eschrich et al., 2003, Altman 1999, Kolen and Hutcheson, 2002, Borgelt and Kruse, 2003), for speeding up clustering have also been proposed.

As seen in the literature, the researchers contributed only to reduce the time complexity or to accelerate the algorithm ; there is no contribution in cluster refinement. In this study, we propose a new algorithm to improve the fuzzy c-means. In this proposed algorithm, an ant colony optimization algorithm is applied to refine the cluster to improve the quality. The paper is organized as follows: section 2 presents the general fuzzy c-means algorithm. Section 3 discusses the proposed cluster refinement algorithm with ant colony optimization. Section 4 presents the results and the work is concluded in section 5.

2. Standard Fuzzy C-Means Clustering

The FCM algorithm, also known as Fuzzy ISODATA, is one of the most frequently used methods in pattern recognition. It is based on minimization of the given objective function to achieve good classifications.

$$J(U,V) = \sum_{i=1}^{n}\sum_{j=1}^{c}(\mu_{ij})^m \parallel x_i - v_j \parallel^2$$

J(U,V) is a squared error clustering criterion, and solutions of minimization of (1) are least-squared error stationary points of J(U,V). The expression, X = $\{x_1, x_2, \ldots, x_n\}$ is a collection of data, where n is the number of data points. V = $\{ v_1, v_2, \ldots, v_c\}$ is a set of corresponding cluster centers in the data set **X**, where c is the number of clusters. μ_{ij} is the membership degree of data x_i to the cluster centre v_j. Meanwhile, μ_{ij} has to satisfy the following conditions:

$$\mu_{ij} \in [0,1], \forall i = 1,...,n, \forall j = 1,...,c$$

$$\sum_{j=1}^{c}\mu_{ij} = 1, \forall i = 1,...,n$$

Where $U = (\mu_{ij})_{n*c}$ is a fuzzy partition matrix, $\parallel x_i - v_j \parallel$ represents the Euclidean distance between x_i and v_j, parameter m is the "fuzziness index" and is used to control the fuzziness of membership of each datum in the range $m \in [1,\infty]$. In this experimentation the value of m= 2.0 was chosen. Although there is no theoretical basis for the optimal selection of m, this has been chosen because the value has been commonly applied within the literature. The FCM algorithm is described in, for example, and can be performed by the following steps:

1. Initialize the cluster centers V = $\{ v_1, v_2, \ldots, v_c\}$, or initialize the membership matrix μ_{ij} with random value and make sure it satisfies the above conditions and then calculate the centers.
2. Calculate the fuzzy membership μ_{ij} using

$$\mu_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{d_{ij}}{d_{ik}}\right)^{\frac{2}{m-1}}}$$

$$where, d_{ij} = \parallel x_i - v_j \parallel, \forall i = 1,...,n,$$

$$\forall j = 1,...c$$

3. Compute the fuzzy centers v_j using

$$v_j = \frac{\sum_{i=1}^{n}(\mu_{ij})^m x_i}{\sum_{i=1}^{n}(\mu_{ij})^m}, \forall j = 1,...,c$$

4. Repeat steps (2) and (3) until the minimum J value is achieved.
5. Finally, defuzzification is necessary to assign each data point to a specific cluster (i.e. by setting a data point to a cluster for which the degree of the membership is maximal).

3. Aco Based Cluster Refinement

Ant-based clustering and sorting was originally introduced for tasks in robotics by Deneubourg et al. (1991). Lumer and Faieta (1994) modified the algorithm to be applicable to numerical data analysis, and it has subsequently been used for data-mining (Lumer and Faieta (1994), graph-partitioning (Kuntz and Snyers 1994, Kuntz and Snyers, 1999, Kuntz et al., 1998) and text-mining (Handl and Meyer 2002, Hoe et al., 2002, Ramos V., and Merelo, 2002).

Such ant-based methods have shown their effectiveness and efficiency in some test cases (Handl et al., 2003). However, the ant-based clustering approach is in general immature and leaves big space for improvements. With these considerations, however, the standard ant-based clustering performs well; the algorithm consists of lot of parameters like pheromone, agent memory, number of agents, number of iterations and cluster retrieval etc. For these parameters more assumptions have been made in the previous works. So far, ants are used to cluster the data points. Here is the first time; we have used ants to refine the clusters. The clusters from the above section are considered as input to this ACO based refinement step.

The basic reason for our refinement is, in any clustering algorithm the obtained clusters will never gives us 100% quality. There will be some errors known as misclustered. That is, a data item can be wrongly clustered. These kinds of errors can be avoided by using our refinement algorithm.

In our proposed method, three ants are used to refine the clusters. These ants are allowed to go for a random walk on the clusters. Whenever it crosses a cluster, it will pick an item from the cluster and drop it into another cluster while moving. And then the quality of the clusters is compared with the drop probability calculated from two cluster validity indexes; Partition Coefficient (PC) and Partition Entropy (PE) are defined as (Bezdek 1981):

$$PC = \frac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{c}\mu_{ij}^{2}$$

$$PE = -\frac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{c}\mu_{ij}\log_{a}(\mu_{ij})$$

PC and PE is used to measure the fuzziness of the fuzzy partition matrix, the lower the fuzziness of a partition is, the larger the PC value (or the smaller the PE value). From these validity indexes the drop probability is calculated as:

$$P_d = PE / PC$$

If P_d is smaller than the previous iteration, then the drop is made permanent and next iteration is continued with the changed cluster indexes. Otherwise, the next iteration is continued with the old cluster indexes.

This random walk is repeated for N number of times. From the following section, it is shown that our refinement algorithm improves the cluster quality. The algorithm is given as:

1. Initialize the cluster centers $V = \{ v_1, v_2,\ldots, v_c\}$, or initialize the membership matrix μ_{ij} with random value and make sure it satisfies the above conditions and then calculate the centers.
2. Calculate the fuzzy membership μ_{ij}
3. Compute the fuzzy centers v_j
4. Repeat steps (2) and (3) until the minimum J value is achieved.
5. Finally, defuzzification is necessary to assign each data point to a specific cluster.
6. Ant based refinement
    a. Input the clusters from fuzzy c-means.
    b. For i = 1 to N do
        i. Let the ants go for a random walk to pick the items
        ii. Drop the items into some other cluster.
        iii. Check whether the quality improving or not by calculating PE and PC.
        iv. If it improves then drop the items permanently.
    c. Repeat

4. Results

Clustering validity is a concept that is used to evaluate the quality of clustering results. If the number of clusters is not known prior to commencing an algorithm, the clustering validity index may be used to find the optimal number of clusters (Rezaee et al., 1998). This can be achieved by evaluating all of the possible clusters with the validity index and then the optimal number of clusters can be determined by selecting the minimum value of the index. Many clusters validation indices have been developed in the past. In the context of fuzzy methods, some of them only use the membership values of a fuzzy cluster of the data, such as the partition coefficient and partition entropy. The advantage of this type of index is that it is easy to compute but it is only useful for the small number of well-separated clusters. Furthermore, it also lacks direct connection to the geometrical properties of the data. In order to overcome this problem Xie and Beni defined a validity index which measures the compactness and separation of clusters (Xie and Beni, 1991). In this paper, the Xie-Beni index has been chosen as the cluster validity measure because it has been shown to be able to detect the correct number of clusters in several experiments (Pal and Bezdek, 1995). Xie-Beni validity is the combination of two functions. The first calculates the compactness of data in the same cluster and the second computes the separateness of data in different clusters. Let S represent the overall validity index, π be the compactness and s be the separation of the fuzzy c partition of the data set. The Xie-Beni validity can now be expressed as:

$$S = \pi / s$$

Where, $$\pi = \frac{\sum_{j=1}^{c}\sum_{i=1}^{n} \mu_{ij}^{2} \| x_i - v_j \|^2}{n}$$

and $$s = (d_{min})^2$$

d_{min} is the minimum distance between cluster centres, given by $\min_{ij} \| v_i - v_j \|$. Smaller values of π indicate that the clusters are more compact and larger values of s indicate the clusters are well separated. Thus a smaller S reflects that the clusters have greater separation from each other and are more compact. The following tables present the results, shows that our proposed method outperforms than the standard method.

Table 1. Performance of Clustering for Wisconsin Breast Cancer Dataset

	Fuzzy C-Means	Refined Fuzzy C-Means with ACO
No. of Classes	2	2
No. of Clusters	2	2
Partition Coefficient	0.8268	0.9861
Partition Entropy	0.2985	0.0392
Xie-Beni Index	3.3862	2.9562

Table 2. Performance of Clustering for Dermatology Dataset

	Fuzzy C-Means	Refined Fuzzy C-Means with ACO
No. of Classes	6	6
No. of Clusters	6	6
Partition Coefficient	0.9433	0.9847
Partition Entropy	0.1412	0.0554
Xie-Beni Index	1.1803	3.4420

5. Conclusion

Cluster analysis is one of the major tasks in various research areas. However, it may be found under different names in different contexts such as unsupervised learning in pattern recognition, taxonomy in biology, partition in graph theory. The clustering aims at identifying and extract significant groups in underlying data. Thus based on a certain clustering criterion the data are grouped so that data points in a cluster are more similar to each other than points in different clusters. Since clustering is applied in many fields, a number of clustering techniques and algorithms have been proposed and are available in literature. In this paper, an ant colony algorithm is presented to improve the cluster from fuzzy c-means clustering. The performance is compared with the standard fuzzy c-means clustering; the result shows the proposed method performs better than the standard method.

REFERENCES

1. Altman, D., 1999. Efficient Fuzzy Clustering of Multi-spectral Images, FUZZ-IEEE.
2. Bensaid, A., J. Bezdek, L.O. Hall, and L.P. Clarke, 1996. Partially Supervised

Clustering for Image Segmentation, Pattern Recognition, V. 29, No. 5, pp. 859-871.

1. Berry, Gordon Linoff, 1996. Data Mining Techniques For marketing, Sales and Customer Support. John Willey & Sons, Inc.
2. Bezdek, J. C., Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, 1981.
3. Bezdeck J.C, Ehrlich R., Full W., 1984. FCM:Fuzzy C-Means Algorithm. Computers and Geoscience 1984.
4. Borgelt, C., and R Kruse, 2003. Speeding Up Fuzzy Clustering with Neural Network Techniques. Fuzzy Systems. V. 2, pp. 852–856.
5. Bradley, P.S., U Fayyad, and C Reina, 1998. Scaling Clustering Algorithms to Large Databases. In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, KDD-1998. pp., 9–15.
6. Canno, R.L., Dave, J.V., Bezdek, J.C., 1986. Efficient Implementation of the Fuzzy C-Means Clustering Algorithm. IEEE Trans. Pattern Anal. Mach. Intel. vol. 8, pp. 248-255.
7. Cheng, T.W., Dmitry B. Goldgof, and Lawrence O. Hall, 1998. Fast Fuzzy clustering. Fuzzy Sets and Systems. V. 93, pp. 49–56.
8. Deneubourg J.L., Goss S., Franks, N. Sendova-Franks A., Detrain C., and Chétien L. 1991. The Dynamics of Collective Sorting: Robot-like Ants and Ant-like Robots. In Proceedings of the 1st International Conference on Simulation of Adaptive Behavior: From Animals to Animats,. MIT Press, Cambridge, MA, USA. Vol. 1, pp. 356-363.
9. Eschrich, S., J Ke, LO. Hall and DB. Goldgof, 2003. Fast Accurate Fuzzy Clustering through Data Reduction. IEEE Transactions on Fuzzy Systems, vol. 11, no. 2, pp. 262–270.
10. Farnstrom, F., J Lewis, and C Elkan, 2000. Scalability for Clustering Algorithms Revisited. ACM SIGKDD Explorations. vol. 2, pp. 51–57.
11. Fayyad, M. U., Piatesky-Shapiro, G., Smuth P., Uthurusamy, R. 1996. Advances in Knowledge Discovery and Data Mining. AAAI Press.
12. Guha, S., R Rastogi, and K Shim, 1998. CURE: An Efficient Clustering Algorithm for Large Databases. In Proceedings of ACM SIGMOD International Conference on Management of Data, pp. 73–84.
13. Guha, S., Rastogi, R., Shim K. 1998. CURE: An Efficient Clustering Algorithm for Large Databases, Published in the Proceedings of the ACM SIGMOD Conference.
14. Gupta, C., and R Grossman, 2004. GenIc: A Single Pass Generalized Incremental Algorithm for Clustering. Proceedings of the Fourth SIAM_ International Conference on Data Mining (SDM 04), pp. 22–24.
15. Handl J., Knowles J., and Dorigo M. 2003. On the Performance of Ant-based Clustering. In Design and Application of Hybrid Intelligent Systems, Frontiers in Artificial Intelligence and Applications,. Netherlands: IOS Press, vol. 104, pp. 204-213.
16. Handl J., and Meyer B. 2002. Improved ant-based clustering and sorting in a document retrieval interface. In Proceedings of the Seventh International Conference on Parallel Problem Solving from Nature, Springer-Verlag, Berlin, Germany, vol. 2439, pp. 913–923.
17. Hathaway, R. J., and JC. Bezdek, 2006. Extending Fuzzy and Probabilistic Clustering to Very Large Data Sets, Journal of Computational Statistics and Data Analysis.
18. Hoe K., Lai W., and Tai T. 2002. Homogeneous ants for web document similarity modeling and categorization. In Proceedings of the Third International Workshop on Ant Algorithms, Springer-Verlag, Heidelberg, Germany, vol. 2463, pp. 256–261.
19. Jain, A.K, Murty MN and Flynn PJ, 1999. Data clustering: a review. ACM Computing Surveys, vol. 31, no.3, pp. 264|323.
20. Kamel, M.S., Selim,S.Z. 1994. New Algorithms for Solving the Fuzzy Clustering Problem. Pattern Recognition. Vol. 27, pp. 421-428.
21. Kolen, J.F., and T Hutcheson, 2002. Reducing the Time Complexity of the Fuzzy C-Means Algorithm. IEEE Transactions on Fuzzy Systems. vol. 10, pp. 263–267.
22. Kuntz P., and Snyers D. 1994. Emergent colonization and graph partitioning. In Proceedings of the Third International

Conference on Simulation of Adaptive Behaviour: From Animals to Animats. MIT Press, Cambridge, MA, vol. 3, pp. 494–500.

23. Kuntz P., and Snyers D. 1999. New results on an ant-based heuristic for highlighting the organization of large graphs. In Proceedings of the 1999 Congress on Evolutionary Computation, IEEE Press, Piscataway, NJ, pp. 1451–1458.
24. Kuntz P., Snyers D., and Layzell P. 1998. A stochastic heuristic for visualizing graph clusters in a bi-dimensional space prior to partitioning. Journal of Heuristics. vol. 5, no. 3, pp. 327–351.
25. Kowalski, G., 1997. Information Retrieval Systems – Theory and Implementation. Kluwer Academic Publishers.
26. Larsen B., and Aone C. 1999. Fast and Effective Text Mining Using Linear-time Document Clustering. KDD-99, San Diego, California.
27. Lumer, E., and Faieta B. 1994. Diversity and adaptation in populations of clustering ants. In Proceedings of the Third International Conference on Simulation of Adaptive Behaviour: From Animals to Animats. MIT Press, Cambridge, MA, vol. 3, pp. 501–508.
28. Lumer, E., and Faieta B. 1995. Exploratory database analysis via self-organization.
29. Ng, R.T., and J Han, 2002. CLARANS: A Method for Clustering Objects for Spatial Data Mining. IEEE Transactions on Knowledge and Data Engineering. Vol. 14, no. 5, pp. 1003–1016.
30. O'Callaghan, L., N. Mishra, A. Meyerson, S. Guha, and R. Motwani, 2002. Streaming-Data Algorithms for High-Quality Clustering, Proceedings of IEEE International Conference on Data Engineering.
31. Pal, N.R., and JC. Bezdek, 1995. On cluster validity for the fuzzy c-means model. IEEE Trans. Fuzzy Systs., Vol. 3, pp. 370-379.
32. Pal, N.R., and JC. Bezdek, 2002. Complexity Reduction for "Large Image" Processing. IEEE Trans on Systems, Man, and Cyber., Part B vol. 32, no. 5, pp. 598–611.
33. Ramakrishnan, Z. R., M Livny, 1996. BIRCH: An Efficient Data Clustering Method for Very Large Databases. Tian ACM SIGMOD International Conference on Management of Data. pp. 103–114.
34. Ramos V., and Merelo JJ. 2002. Self-organized stigmergic document maps: Environments as a mechanism for context learning. In Proceedings of the First Spanish Conference on Evolutionary and Bio-Inspired Algorithms, Centro Univ. M´erida, M´erida, Spain, pp. 284–293.
35. Rezaee, M.R., BPF. Leieveldt, and JHC. Reiber, 1998. A New Cluster Validity Index for the Fuzzy C-Means. Pattern Recognition Letters. Vol. 19, pp. 237-246.
36. Theodoridis, S., Koutroubas, K. 1999. Pattern recognition, Academic Press.
37. Wang, X., J M. Garibaldi, 2005. Simulated Annealing Fuzzy Clustering in Cancer Diagnosis. Informatica, vol. 29, pp. 61–70.
38. Wang, X., Y Wang, L Wang, 2004. Improving fuzzy c-means clustering based on feature-weight learning. Pattern Recognition Letters. vol. 25, pp. 1123–1132.
39. Xie, X. L., and G. Beni, 1991. A validity measure for fuzzy clustering. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 13, pp. 841-847.