International Journal of Engineering & Technology, 7 (3.33) (2018) 131-133

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

A Comparative Study between of Fuzzy C-Means Algorithms and Density based Spatial Clustering of Applications with Noise

Kwang Kyu Lee*

**Shinhan University, Seoul, Korea*
**Corresponding author E-mail: kklee@shinhan.ac.kr*

Abstract

Data mining technology has emerged as a means of identifying patterns and trends from large amounts of data and is a computing intelligence area that provides tools for data analysis, new knowledge discovery, and autonomous decision making. Data clustering is an important problem in many areas. Fuzzy C-Means(FCM)[11,12,13] is a very important clustering technique based on fuzzy logic. DBSCAN(Density Based Spatial Clustering of Applications with Noise)[8] is a density-based clustering algorithm that is suitable for dealing with spatial data including noise and is a collection of arbitrary shapes and sizes. In this paper, we compare and analyze the performance of Fuzzy C-Means and DBSCAN algorithms in different data sets.

Keywords: *Data Clustering Algorithm, Data Mining, DBSCAN, FCM, Fuzzy C-Means, K-means*

1. Introduction

Data mining, which processes and analyzes large amounts of data, is somewhat difficult, but data mining is a very important and useful tool in the software field [2]. Data mining is the process of automatically discovering useful information in a large data store. Data mining techniques are applied to explore new and useful patterns that can be traversed across large databases and noticed [7]. Clusters used in data mining are data description methods that are used as common techniques for data analysis in various fields, such as machine learning, pattern recognition, image analysis, and bioinformatics. Cluster analysis is a grouping of data objects based on information found in the data describing the objects and their relationships. The goal is to increase the similarity of the objects belonging to the same group and to make a difference with the objects of other groups. Clustering becomes more evident as the similarities among the objects in the group and the differences between the groups are increased [5]. In this paper, we analyze FCM, which is a cluster method to update the center point after initialization, and density-based DBSCAN algorithm, which grasps regions with high density by low density region using various data sets.

2. Fuzzy C-Means (FCM)

Generally, fuzzy clusters are used as an unsupervised learning strategy for grouping data, but they are also useful for generating fuzzy rules (if ~ then ~) from data [12]. The structure of the fuzzy rules depends on the nature of the data used. For example, in fault diagnosis or pattern classification, a fuzzy rule is created to determine what data should be categorized, and fuzzy control, system recognition, or function approximation is designed to describe the continuous relationship between input and output variables. In image analysis and recognition, it is used to detect and separate spatial geometries such as circles and ellipses, and this is called a shell clustering algorithm [13]. The purpose of a fuzzy cluster is to partition any data set into a specific number of fuzzy clusters. The most widely used fuzzy clustering method to date is the FCM algorithm. The algorithm consists of a set of n items X = {X1,X2,...,Xn} is divided into c fuzzy clusters, the objective is to find a fuzzy partition F^={(F_1)^,(F_2)^,⋯,(F_c)^}that minimizes the objective function of Eq. (2.1).There is a difference between the values of the parameter m and the fuzzy division matrix for the hard cluster. The parameter m that represents the fuzziness of a partition can have a value greater than 1, the larger the value, the higher the ambiguity (m is 1 indicating a clear hard cluster that is not ambiguous). The value of m is generally known to provide good results of 1.25 or 2. However, it is possible to choose a value of m that is appropriate for the application [9]. μik denotes the degree to which data Xk belongs to the fuzzy cluster, and the element of (c × n) size fuzzy split matrix U = [μik] satisfies the condition of Eq. (2.1). The sum of the degree to which specific data belongs to all the clusters is equal to 1 is the same as in the case of hard clusters.

$$J_m(U,V:X) = \sum_{i=1}^{c}\sum_{k=1}^{n}(\mu_{ik})^m \|X_k - V_i\|^2 \quad (2.1)$$

In (2.1), V = (V1, V2,…,Vc) is the set of center vectors of c clusters. That is, Vi is the center vector of the i-th cluster. |Xk-Vi| represents the geometric distance between data Xk and the center of the i-th cluster. If the dimension of V and Xk is p, Vi = (Vi1, Vi2,…,Vip) and Xk = (Xk1,Xk2,...,Xkp). The parameter m is a value that adjusts the degree to which the data belongs to the cluster. It is 1 for hard clusters. μik denotes the degree to which the data Xk belongs to the cluster (F_i)^and has a value of 0 or 1 and satisfies the condition of (2.2) with an element of the (c × n) sized partitioning matrix U = [μij]. Data must belong to only one cluster, since the sum of the degrees of certain data belonging to all the clusters must be 1.

$\mu_{ik} \in \{0,1\}, \quad \sum_{i=1}^{c} \mu_{ik} 1$ (2.2)

The FCM clustering algorithm is summarized as follows.

① Select the number of divisions $c(2 \le c \le n)$ and m.
② The initial value of the fuzzy division matrix $U^{(t)}$ is determined. A random value satisfying the expression (2.2) is used.
③ Calculate the centroid V of the cluster using equation (2.3).

$$V_i^{(t+1)} = \frac{\sum_{i=1}^{c}(\mu_{ik}^{(t)})^m X_k}{\sum_{i=1}^{c} \mu_{ik}^{(t)}} \quad m > 1 \;\; i = 1, \cdots, c \tag{2.3}$$

④the fuzzy partition matrix is updated using equation (2.4).

$$\mu_{ik} = \frac{1}{\sum_j^c \left(\frac{|X_k - V_i|^2}{|X_k - V_j|^2} \right)^{1/(m-1)}} \; \mathrm{i} = 1, \cdots, \mathrm{c} \quad \mathrm{k} = 1, \cdots, \mathrm{n} \tag{2.4}$$

⑤ IF |U^((t+1))-U^((t)) |<δ is satisfied, the process is terminated. Otherwise, the process returns to step ③ and the process is repeated. Normally 10-3 is used as the δ value.

3. Density based Algorithm (DBSCAN)

DBSCAN is a clustering algorithm based on [2,7] density, suitable for handling spatial data including noise, and can distinguish clusters of various shapes and sizes. The cluster and noise are intuitively classified based on the density of the points. To do this, we define a few

[Definition 1] The Eps-neighborhood of a point p is a set of neighborhoods within a radius Eps from p. That is, it is the Eps-neighborhood of the

point p. i.e. NEps(p) = {q∈ D|dist(p, q) ≤Eps}

[Definition 2] The fact that one point p is directly density-reachable from point q

• p must be in the neighborhood of q, p∈ NEps(q)

• [NEps(q)] ≥MinPts (core point condition): q is a core point. That is, having enough neighbors.

[Definition 3] The fact that one point p can arrive at the density from point q means that there is a direct connectable density connection from

pi+1 to pi from p to q.

[Definition 4] The fact that one point p is density-connected from point q means that there exists a point o that can arrive at the density from

p and q.

[Definition 5] Cluster

Let D be a dataset of points, C denote clusters, and if Eps and MinPts are a subset of D, not an empty set, then:

• It is q for all points p, q and p∈C is q if density is available from q∈C

• Every p, q, p∈C : p is said to be densely connected to q.

[Definition 6] Noise

When Ci is a cluster in database D, the noise is that it does not belong to any clusters in D. That is, Noise ={ ├ p∈D┤ | ∀ i:p∉C_i }

Each cluster is defined as the largest set of dense-connected points, and each point in a cluster has a minimum number of neighbors of MinPts or greater in a given radius Eps. The time complexity of DBSCAN is O(n*logn) [5]. Figure 1 shows a database 1 with four spherical clusters of different sizes, a database 2 with non-convex clusters, a database 3 with clusters of different shapes and sizes and a database 3 with noise. DBSCAN was applied The results of each cluster are shown.

Figure 1: Cluster result when DBSCAN is applied

4. Experimental Results

An efficient implementation of MATLAB between DBSCAN and FCM was implemented in the Iris dataset of the well-known UCI machine learning store. The data consisted of 150 samples, each consisting of 50 classes (setosa, versicolor, virginica) [13]. Experimental Results DBSCAN time complexity is O(n*logn), which enables efficient search of all points within a given distance based on one specific point because it has a data structure like kd-trees in low dimensional space [5] , And the time complexity of FCM is O(n2) [11]. Table 1 shows the time complexity of FCM and DBSCAN when the number of clusters is changed from 1 to 4. Figure 2 shows a two-dimensional graph based on Table 1.

Table 1: Time Complexity when Number of cluster varying

Sequence Number	Number of cluster	FCM	DBSCAN
1	1	2500	2500
2	2	10000	5918
3	3	22500	9647
4	4	40000	13608

Table 2 shows the elapsed time of FCM and DBSCAN for different number of repetitions. DBSCAN can handle clusters with various sizes and shapes, and yields the same results until 10 iterations, but FCM yields slightly different results from one to five iterations, the same result could be calculated. Figure 2 is a graph based on Table 2.

Table 2: Time Complexity when Number of Iterations varying

Sequence Number	Number of Iteration	FCM	DBSCAN
1	1	2500	2500
2	5	10000	5918
3	10	22500	9647

As a result, we found that FCM requires more computation time than DBSCAN because we have to calculate weights according to the information that all data belongs to.

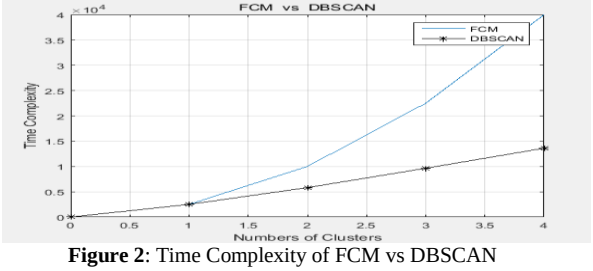

Figure 2: Time Complexity of FCM vs DBSCAN

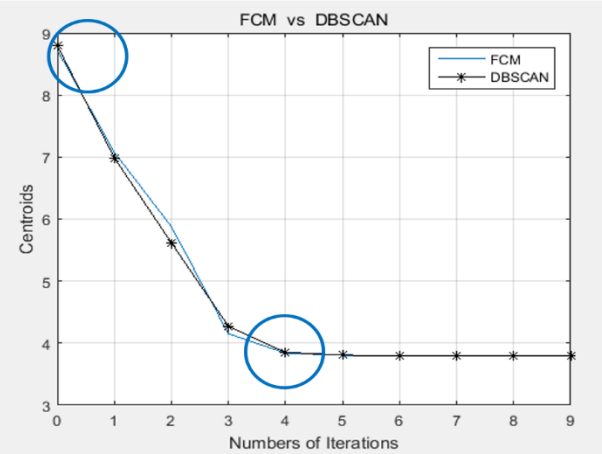

Figure 3: Time Complexity when Number of Iterations varying

5. Conclusion and Future Research

Choosing a particular clustering algorithm depends on the type of data to be clustered and the purpose of the clustering application. The FCM is suitable for dealing with problems related to understanding of patterns or noise data, mixed media information, and human interactions, and can provide approximate solutions faster. They were mainly used to find association rules and functional dependencies and image retrieval. Also, under certain cases, we can't be certain that data belongs to only one cluster. This is because some data attributes can belong to more than one community. On the other hand, since the DBSCAN algorithm uses the density-based definition of the cluster, it can extract irregular shape and size clusters that are relatively noise-robust and can't be handled using k-means. In this paper, we compare and analyze FCM and DBSCAN algorithm of Iris data set. Based on the experiments, we found that the computation time of the DBSCAN algorithm is less than FCM. FCM requires more computation time than DBSCAN because the center point of all data is calculated and weighted according to the information to which data belongs. This study can conclude that the time complexity of DBSCAN is better than FCM. In order to solve this problem, the time-complexity of O(n2) can be improved by using the improved OPTICS (Ordering Points to Identify the Clustering Structure) algorithm .

References

[1] A. Asuncion and D. J. Newman, UCI Machine Learning Repository Irvine, CA: University of California, School of Information and Computer Science, 2013.
[2] A. K. Jain, M. N. Murty and P. J. Flynn, "Data Clustering: A review", ACM Computing Surveys, vol. 31, no. 3, 1999.
[3] A. Rui and J. M. C. Sousa, "Comparison of fuzzy clustering algorithms for Classification", International Symposium on Evolving Fuzzy Systems, 2006 , pp. 112-117.
[4] B. Jeon, Y. Yung and K. Hong "Image segmentation by unsupervised sparse clustering, " pattern recognition letters 27science direct,(2006) 1650-1664
[5] https://en.wikipedia.org/wiki/DBSCAN#Complexity
[6] H.P.K and l.M.P, "Density-Based Clustering of Uncertain Data", KDD'05, August21-24, 2005, Chicago, Illinois, USA.
[7] J. C. Bezdek ,"Pattern Recognition with Fuzzy Objective Function Algoritms", Plenum Press, New York, 1981
[8] M. Ester, H. Kriegel, J. Sander, X. Xu, ""A Density-Based Algorithm for Discovering
[9] Clusters in Large Spatial Databases with Noise"", Proc. of ACM SIGMOD 3rd International
[10] Conference on Knowledge Discovery and Data Mining, pp. 226-231, AAAI Press, 1996.
[11] Richard J. Hathaway and James C. Bezdek, Extending Fuzzy and Probabilistic Clustering to Very Large Data Sets, Journal of Computational Statistics and Data Analysis, 2006, accepted.
[12] S. l Har-Peled and B. Sadri, "How fast is the k-means Method," in ACM-SIAM Symposium on Discrete Algorithms,Vancouver, 2005.
[13] Soumi Ghosh and Sanjay Kumar Dubey, "Comparative Analysis of K-Means and Fuzzy C-Means Algorithms", IJACSA, Vol. 4, No.4, 2013,pp. 35-39.