Dempster-Shafer Theory of Evidence in Single Pass Fuzzy C Means

Alireza Chakeri, Iman Nekooimehr *, Lawrence O.Hall, Fellow, IEEE
Department of Computer Science and Engineering
*Department of Industrial and Management Systems
University of South Florida
Tampa, Florida
Email: chakeri@mail.usf.edu, hall@cse.usf.edu, nekoeimehr@mail.usf.edu

Abstract—Clustering large data sets has become very important as the amount of available unlabeled data increases. Single Pass Fuzzy C Means (SPFCM) is useful when memory is too limited to load the whole data set. The main idea is to divide dataset into several chunks and to apply FCM to each chunk. SPFCM uses the weighted cluster centers of the previous chunk in the next chunks. Although when the number of chunks is increased, the algorithm shows sensitivity to the order the data processed. Hence, we improved SPFCM by recognizing boundary and noisy data in each chunk and using it to influence clustering in the next chunks. In this regard, the proposed approach transfers the boundary and noisy data as well as the weighted cluster centers to the next chunks. We show that our proposed approach is significantly less sensitive to the order in which the data is loaded in each chunk.

Keywords—fuzzy c means (FCM) clustering, single pass FCM, evidential FCM, belief functions, boundary data.

I. INTRODUCTION

Clustering is a data analysis approach aiming to group the number of objects into different clusters such that objects in a cluster have more similarity to one another than the others. Using hard partitioning methods, an object only belongs to one cluster and it cannot be included in another cluster. On the other hand, in fuzzy clustering each object may belong to more than one cluster having different degrees of membership such that the memberships sum to one. The most popular fuzzy partitioning method is Fuzzy C-means (FCM) clustering [1]. However, FCM may have poor robustness against noise and outliers. A brief survey of variations of FCM and their application in image processing can be found in [2-3]. Some approaches have been proposed to relax the normalization constraint in FCM. Consequently, the resulting clusters are referred to as possibilistic. Krishnapuram et al in [4] introduced the possibilistic clustering algorithm (PCM) by modifying the objective function of FCM. Also rough C-means and fuzzy rough C-means were proposed in [5] by introducing the concept of lower and upper approximations of clusters. In [6] an approach to cluster noise objects poorly fit by the existing clusters was proposed. In addition, to gain a deeper insight into the data and improve robustness with respect to outliers, the credal partition was introduced in [7] based on the Dempster-Shafer theory of belief functions. Dempster-Shafer theory (DST) [8-9] can be considered as a generalization of probability theory. The representation of uncertainties in DST is similar to that in conventional probability, however, DST has one significant new feature. It allows the probability to be assigned to subsets besides the individual elements of the Universe of Discourse. This new feature makes the theory potentially attractive and useful for representing the amount of probability or belief which can be committed to a subset, but nothing smaller. This reflects the imprecision of the evidence by indicating that through lack of information this probability value can not be further subdivided among the elements in that subset. Hence, in [7], a credal partition was defined as a generalization of the hard, fuzzy and possibilistic partition. In this regard, based on an algorithm called ECM [10], a degree of membership is assigned for each object not only to single clusters, but also to any subsets of the set of clusters.

Applying a clustering algorithm to large data sets becomes very important issue as the number of available large datasets increase. Some efforts have been made to apply clustering algorithms to large data sets [11-12]. They are mainly based on sampling the datasets [11] or loading partial data incrementally [12]. In [13], an approach to make FCM faster for large data sets was introduced based on sampling from data. It initialized the centroids for FCM using cluster centers from a limited random sampling from data. Also, in [14] a data reduction approach combined similar examples into weighted examples. On the other hand, recently a modified FCM algorithm was introduced for extremely large datasets when the limitation of memory has to be considered. Hore et al in [15-16] introduced a single pass FCM (SPFCM) and online FCM (OFCM) which do not need any complicated data compression technique. The SPFCM approach produces final partitions very similar to FCM by a single pass through the data. The proposed method [15] is based on partitioning the data set into several sub partitions and then clustering each of them. In this regard, after clustering the first chunk using
fuzzy C-means, data of the second chunk and previous cluster centroids as represented as weighted data are used when clustering the second chunk, and so on. They showed that SPFCM has a significant speed-up compared to fuzzy c-means. The convergence proofs for SPFCM and OFCM were also given in [17].

When the number of chunks increases in SPFCM, i.e. the number of data in each chunk decreases, the extrema distribution obtaining from the reformulated optimization function [18] may not be uniform. This can be clearly observed over experiments with different orderings of the data. In fact, it is sensitive to the initialization of data in each chunk. Hence, in this paper, we introduce an improvement to SPFCM called SPECM by using the boundary and noisy data. Based on this improvement, the boundary and noisy data in each chunk are identified by the ECM algorithm and their centroids are transferred to the next chunk as well as cluster centers. The idea of modification is that the boundary and noisy data may belong to the clusters in the next chunks, but SPFCM assigns those data to the cluster centers of the current chunk with almost equal degrees of membership. Hence, using SPECM, the boundary and noisy data have a chance to be put in appropriate clusters in the next chunks by including representations of them in the next chunks as well as cluster centers. The proposed approach will provide approximately similar clustering quality by loading a very small percentage of the data in each chunk compared to clustering all data at once using FCM.

II. SING E PASS EVIDENTIAL C-MEANS ALGORITHM

In this section, we will briefly explain Evidential C Means (ECM) and then the single pass algorithm and how these two concepts combine for clustering super large data sets. ECM is the application of belief function theory in c means clustering [10]. In ECM, an instance not only can belong to each cluster with a membership value, it can also belong to a subset of clusters with some degree, i.e. power set of the set of all clusters. Let’s assume that \( \Omega \) is the set of all possible clusters, that is \( \Omega = \{w_1, w_2, \ldots, w_l\} \) with c clusters, the boundaries are defined as the subsets of \( \Omega \) with two or more members (e.g., \( A_1 = \{w_1, w_2\} \), \( A_2 = \{w_1, w_2, w_3\} \)). Using ECM, we can find the membership value of each instance belonging to each cluster or to the boundaries of the clusters. Also it defines the outliers as the instances which belong to the empty subset.

The single pass fuzzy C-means (SPFCM) algorithm was developed to cluster very large data sets fast [15]. It divides large data in smaller partitions called partial data accesses (PDA) based on the available memory allocation, assuming that the data is randomly scrambled on the disk. After scanning the first PDA, the algorithm divides the instances into c clusters using ECM and then the centers of the clusters are weighted and sent to be clustered again with new points loaded in the next PDA. In other words, instances in the first PDA are condensed into c weighted points. These weighted points are the centers of the clusters such that, for the first PDA, their weights are calculated by summing the membership values of instances in them. It should be mentioned that the weights for the first PDA will sum up to the number of instances within the PDA, \( n_1 \). For the second PDA up to the last one, the weighted points and new loaded points are clustered again to form c new clusters, and then they will be weighted and sent to be clustered with the next PDA until all the examples will be scanned once. Note that the weights for the second PDA will sum up to the number of instances scanned in the first PDA and second PDA, \( n_1 + n_2 \).

For the third PDA it will be \( n_1 + n_2 + n_3 \), and for the last PDA it will be the cardinality of the original dataset.

Our proposed algorithm works pretty much the same as SPFCM except that it uses ECM instead of FCM to cluster the instances and finds the boundary and noisy instances as well as cluster centers in each PDA. Our algorithm not only sends weighted cluster centers to the next PDA, it also sends boundary point weighted centers to the next chunk of data. Using ECM to cluster instances in each PDA will also help us to detect outliers, which will be instances with a tendency to belong to empty set. The outliers will also be transferred to the next PDA.

Assume we have a very large data set of size \( N \) which is more than memory size. To solve this problem our algorithm randomly chooses \( n < N \) instances out of the whole data set each time without replacement. For the first \( n \) instances, the proposed algorithm uses ECM to find the centers of clusters, the centers of boundary and noisy points and the membership value that each instance belongs to each cluster or the borders of the clusters. Then the centers of clusters, the noisy data point, and the centers of boundary points are weighted and added to the next \( n \) points to be clustered again using ECM. The centers of clusters, the noisy data points and centers of boundary points will be detected again and will be sent to the next random subset of overall data. This process will be repeated until all \( N \) points are scanned. However, because it is not necessary to identify the boundary and noisy data in the last PDA, we use FCM in the last PDA to find the final cluster centers. The weights for the first \( n \) instances are calculated by the summation of membership values of examples in a cluster or boundary set. For subsets after the first one, the weights are calculated by taking into account the previous PDA weights.

A. ECM Details

Let \( \{x_1, x_2, \ldots, x_n\} \) be the set of data points in \( \mathbb{R}^p \) and \( c \) be the number of clusters. Let matrix \( V \) of size \( c \times p \) represents the cluster centers. FCM tries to find a fuzzy membership matrix \( U \) of size \( c \times n \) by minimizing the following objective function

\[
J_m(U,V) = \sum_{i=1}^{c} \sum_{k=1}^{n} u^m_{ik} D_{ik}(x_k, v_i) .
\]
In order to be minimized, \( U \) and \( V \) should be calculated as

\[
\begin{align*}
    u_{ik} &= \frac{D_{ik}(x_k, v_i)}{\sum_{j=1}^{c} D_{ik}(x_k, v_j)^{1-m}} \\
    v_i &= \sum_{j=1}^{n} \left( u_{ij} \right)^m x_j
\end{align*}
\]

where \( m \) is any number greater than 1, \( u_{ik} \) is the membership value of the \( i \)th instance belonging to the \( k \)th cluster, \( v_i \) is the center of the \( i \)th cluster, \( n \) is the number of examples and \( c \) is the number of clusters. \( D_{ik}(x_k, v_i) \) can be any distance measure between the \( k \)th instance and the \( i \)th cluster center, such as the Euclidean distance. FCM can be solved through an iterative optimization structure, but it should be calculated as

\[
\sum_{j=1}^{n} w_{ij} v_j
\]

subject to

\[
\sum_{j=1}^{n} u_{ij}^m + u_{jk}^m = 1
\]

where the term \( c_j^m \) was added to penalize subsets in \( \Omega \) with large cardinality and can be controlled by \( \alpha \). The term \( \delta^2 \) was also added to the second part to account for the amount of data considered as outliers. Note that in the first part an instance can belong to any subset of \( \Omega \) except \( \phi \). As we see the empty set was treated differently in this method by removing it from the first part of the objective function.

To minimize the objective function, in [10] they have used the Lagrangian multiplier and calculated \( U \) as follows

\[
\begin{align*}
    u_{ik} &= \frac{-c_j^m D_{ik}(x_k, v_i)^{-2/m} - c_j^{m-2} u_{ik}^m}{\sum_{j \neq \phi, \phi \in \Omega} \left( c_j^{m-2} D_{ik}(x_k, v_j)^{-2/m} + \delta^2 \right)^{-1/m}} \\
    v_i &= \sum_{j=1}^{n} \left( u_{ij} \right)^m x_j
\end{align*}
\]

For calculating \( V \), [10] suggests to first find \( B \) and \( H \) as

\[
B_n = \sum_{i=1}^{n} x_{ij} \sum_{r \in \Omega \setminus \{ \emptyset \}} c_r^{\alpha r - 1} u_{ir}^m \quad \forall r \in [1, c], \forall t \in [1, p]
\]

where \( x_{ij} \) is the \( r \)th dimension of the \( i \)th example, \( w_r \in A_j \) refers to any subset of \( \Omega \) which contains the \( r \)th cluster and \( p \) is the dimension of data points in the dataset.

\[
H_m = \sum_{i=1}^{n} \sum_{r \neq \{ \emptyset \}} \sum_{t \in [1, c]} c_r^{\alpha r - 2} u_{ir}^m \quad \forall r, i \in [1, c]
\]

where \( \{ j | \{ w_r, w_l \} \subset A_j \} \) refers to any subset of \( \Omega \) which contains the \( r \)th cluster and the \( k \)th cluster at the same time. After computing \( B \) and \( H \), \( V \) can be calculated as the solution of the following linear equation system

\[
HV = B
\]

Note that \( V \) denotes the centers of clusters not the boundary centers which consist of subsets with 2 or more members. For finding their centers we need to use the following formulae

\[
s_{k,j} = \begin{cases} 
1 & \text{if } w_k \in A_j \\
0 & \text{otherwise}
\end{cases}
\]

\[
v_j = \frac{1}{c_j} \sum_{k=1}^{c} s_{k,j} v_k
\]

in which \( v_j \) is associated with \( A_j \), the centroid of the \( j \)th subset and \( c_j = |A_j| \) refers to the cardinality of \( A_j \). The algorithm terminates when \( \| V_{t+1} - V_t \| < \epsilon \) that is, in two consecutive iterations the center of clusters don’t change more than \( \epsilon \).

### A. Weighted ECM

As we mentioned before, our algorithm uses ECM in each PDA to find the centers of \( A_j \) and then, it weighs them and sends them to be clustered again along with the new instances scanned in the next PDA. ECM needs to be modified for weighted points. Therefore we modified ECM objective function as

\[
J_m(u, v) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_j^{\alpha j - 1} u_{ij}^m D_{ij}(x_i, v_j) + \sum_{k=1}^{n} w_k \delta^2 u_{jk}^m
\]

subject to

\[
\forall i \in \Omega, A_i \neq \emptyset, \forall k \in [1, n]
\]

and

\[
u_{jk} = 1 - \sum_{i \neq j \in \Omega} u_{ik} \quad \forall k \in [1, n].
\]
\[
\sum_{i | A_i \neq \emptyset} u_{ik}^m + u_{\hat{k}k} = 1 .
\] (14)

Also \( H \) and \( B \) were adjusted as

\[
B_{rt} = \sum_{i=1}^{n} x_i w_i \sum c_j^{-1} u_{ij}^m \forall r \in [1, c], \forall t \in [1, p] \tag{15}
\]

\[
H_{rl} = \sum r_i \sum c_j^{-2} u_{ij}^m \forall r, l \in [1, c] . \tag{16}
\]

Finally, \( U \) after adjustment remains unchanged as

\[
u_{ik} = \frac{-c_i / m^{-1} D_{ik}(x_k, v_j)^{-2/m} - \sum_{j'} c_{j'}^{-2} D_{jk}(x_k, v_{j'})^{-2/m} + \delta^{-2/m}}{\sum_{j' \in A_j} c_{j'}^{-2} D_{jk}(x_k, v_{j'})^{-2/m} + \delta^{-2/m}} \tag{17}
\]

\( \forall i \mid A_i \subseteq \Omega, A_i \neq \emptyset, \forall k \in [1, n] \).

B. Calculation of Weighted Points

For the first PDA, we only have \( n \) instances \( \{x_1, x_2, \ldots, x_n\} \) and there will be no previous weighted points. Thus the algorithm uses regular ECM to find the centroids \( v_j \), \( 1 \leq i \leq 2^c \) and the membership values \( u_{ij} \), where \( 1 \leq i \leq 2^c \) and \( 1 \leq j \leq n \). The weights of centroids in the first PDA are calculated such that

\[
w_i = \sum_{j=1}^{n} (u_{ij}) w_j \forall i \in [1, 2^c - 1] \tag{18}
\]

where \( w_j = 1, \forall 1 \leq j \leq n \). \( i \) is considered less than or equal to \( 2^c - 1 \), since the last subset corresponds to the empty subset which is considered to be outliers. Assume that the set \( O_k = \{x_{k1}, \ldots, x_{kp}\} \) is set of noisy data in the \( k \)th chunk, and

\[
U_{\phi k} = \{u_{\phi k1}, \ldots, u_{\phi kp}\} \] is the set of degrees of belonging of the noisy data to the set \( \phi \) in the \( k \)th chunk. For the second PDA, the weights for \( 2^c - 1 \) centers and noisy data will be

\[
w_i = w_i, \quad \forall i \in [1, 2^c - 1] .
\]

\[
w_i = u_{\phi i1}, \quad \forall i \in [2^c - 1 + |\Omega_1|] .
\] (19)

Also the weights for the \( n \) new instances just uploaded will be

\[
w_i = 1, \quad \forall i \in [2^c + |\Omega_1|, 2^c - 1 + |\Omega_1| + n] .
\] (20)

The weights of centroids for the rest of PDA are calculated the same as equation in (18).

IV. MATERIALS AND EXPERIMENTAL DESIGN

Two small data sets, i.e., Iris and Pima, were used in this study to test the developed SPECM algorithm. The Iris plant data set consists of 50 samples from each of three species of Iris, i.e., Iris setosa, Iris virginica and Iris versicolor. Four numeric attributes (the length and the width of the sepal and petal) were measured from each sample. The visualization of the Iris data set indicates that one class is completely separable and the other two classes intersect each other. In this study, the number of clusters for Iris data set was set as 3. The Pima Indians Diabetes database consists of 8 medical measurements (e.g., Diastolic blood pressure, Diabetes pedigree function and Triceps skin fold thickness) of 768 subjects. A binary class variable is also provided in the data set to indicate whether the subject has diabetes or not.

First, the data was randomized and the weights were initialized to one. The data samples were then divided into multiple chunks and sequentially provided to the SPECM model developed here. For the Iris data, each chunk consists of 10 data samples and there are 15 chunks in total. For the Pima data set, on the other hand, 16 chunks were created and there were 48 data in each chunk. Furthermore, for each data set, the experiment was repeated 50 times, each time a new randomization of the data was done. It is noted that in the experiment, we increased the exponent for cardinality \( p \) (see equation 13) as more PDAs were scanned. In this way, there would be fewer data samples on the boundaries of clusters as more instances are scanned. To illustrate the model performance, the reformulated optimization criteria \( R_m \) was computed as

\[
R_m(V) = \sum_{k=1}^{n} \left( \sum_{i=1}^{c} D_{ik}(x_k, v_j)^{1/m} \right)^{1/(1-m)} \tag{21}
\]

V. EXPERIMENTAL RESULTS

Two features, i.e., petal length and petal width, are used for the visualizing a partition of the Iris data. The partition is visualized in Fig. 1. As discussed above, just 10 data samples form a chunk and are clustered by the SPECM model. It is noted that the separable class (shown with ‘+’ symbol) is well separated on the lower-left corner and the two overlapped classes (‘*’ and ‘.’ symbols) are also separated effectively on the upper-right corner. In the SPFCM approach, on the Iris data set, the two overlapped classes came out as a single cluster and the separable one got split into 2 clusters [16] some times. This has not happened in our experiments (see Fig. 1). As a result, the average error of assigning classes to data points over 50 experiments in SPFCM is 15.6, but using SPECM the average error over 50 experiments is 14.7. This is
because the developed method models the data more completely than SPFCM, and hence is less sensitive to the order in which the data is loaded in each chunk. In each new chunk, the weighted cluster centroids as well as the data samples on the cluster boundaries are incorporated. Thus, the developed SPECM algorithm not only maintains the basic function of SPFCM, which enables partitioning a large amount of data accurately and quickly, but also eliminates the sensitivity to the loading of data samples. We extensively tested our algorithm using some small public data sets and both results show that the SPECM is more robust in terms of extrema obtained. The proposed approach can handle data with different number of clusters, not limited to the 3 clusters Iris data set and 2 clusters Pima data set. The developed SPECM algorithm has great potential in the fuzzy clustering of large data sets, e.g. streaming data.

VI. CONCLUSION

In this study, the SPECM algorithm was developed and tested for different data sets. The developed algorithm improves on SPFCM by incorporating cluster boundary and noisy data in each chunk. Fig. 1. The three clusters are represented with symbols ‘+’, ‘·’, ‘*’.

The extrema distributions of FCM and SPECM on the Pima data over 50 experiments are shown in Figure 2 such that 48 data samples are in one chunk and in total 16 chunks are clustered sequentially by SPECM. The x-axis is the number of the experiment and y-axis is the $R_{\text{in}}$ value. It can be observed that for the extrema of SPECM, they are distributed closely around 3.46 and no poor partition is found. The figure also shows that SPECM can cluster Pima data set better than FCM in terms of $R_{\text{in}}$ value equal to -2.88 percent, while SPFCM performs 2.22 percent worse than FCM.

Fig. 2. Extrema distribution of FCM and SPECM on the Pima data set.

REFERENCES