EXTENSION OF THE FUZZY C MEANS CLUSTERING ALGORITHM TO FIT WITH THE COMPOSITE GRAPH MODEL FOR WEB DOCUMENT REPRESENTATION

Mr. Kaushik K. Phukon MCA, *Department of Computer Science, Gauhati University, Guwahati- 781014, Assam, India.*
E-mail:kaushikphukon@gmail.com
Prof. Hemanta K. Baruah, *Vice Chancellor, Bodoland University, Kokrajhar-783370, Assam, India.*
E-mail:hemanta_bh@yahoo.com

UDK: 004.738.12:519.178

Abstract: Clustering techniques are mostly unsupervised methods that can be used to organize data into groups based on similarities among the individual data items. Fuzzy c-means (FCM) clustering is one of well known unsupervised clustering techniques, which can also be used for unsupervised web document clustering. In this chapter we will introduce a modified method of clustering where the data to be clustered will be represented by graphs instead of vectors or other models. Specifically, we will extend the classical FCM clustering algorithm to work with graphs that represent web documents (Phukon, K. K. (2012), Zadeh, L. A. (1965). Dunn, J. C.(1974)). We wish to use graphs because they can allow us to retain information which is often discarded in simpler models.

Keywords: Graph, Web Document, Hard Partition, Fuzzy Partition, Fuzzy C- Means.

1. INTRODUCTION

Fuzzy clustering is well-known not only in fuzzy community, but also in the related fields of data analysis, neural networks, and other areas in computational intelligence. The FCM algorithm, proposed by Dunn, J. C. (1974) and extended by Bezdek, J. C. (1981), Cannon, R. L., Dave, J. V., Bezdek, J. C. (1986) can be applied if the objects of interest are represented as points in a multi-dimensional space. FCM relates the concept of object similarity to spatial closeness and finds cluster centers as prototypes. Several examples of application of FCM to real clustering problems have proved the good characteristics of this algorithm with respect to stability and partition quality.

In general, cluster analysis refers to a broad spectrum of methods which try to subdivide a data set X into c subsets (clusters) which are pair wise disjoint, all nonempty, and reproduce X via union. The clusters then are termed a hard (i.e., non-fuzzy) c-partition of X. A significant fact about this type of algorithm is the defect in the underlying axiomatic model that each point in X is unequivocally grouped with other members of its cluster, and thus bears no apparent similarity to other members of X. One such manner to characterize an individual point's similarity to all the clusters was introduced in 1965 by Zadeh. The key to Zadeh's idea (Zadeh, L. A. (1965)) is to represent the similarity a point shares with each cluster with a function (termed the membership function) whose values (called memberships) are between zero and one. Baruah (2011) has defined the membership function of a normal fuzzy number N= $[\alpha, \beta, \gamma]$ as

$$\mu_N(x) = \begin{cases} \Phi_1(x) & \text{if } \alpha \le x \le \beta, \\ \Phi_2(x) & \text{if } \beta \le x \le \gamma, \\ 0 & \text{otherwise.} \end{cases}$$

(Eq: 1.1)

Where $\Phi_1(x)$ and $(1-\Phi_2(x))$ are two independent distribution functions defined in $[\alpha,\beta]$ and $[\beta,\gamma]$ respectively.

Clustering techniques are generally applied to data that are quantitative (numerical), qualitative (categorical), or a mixture of both. But in this chapter we are going to put forward a means for clustering graphical objects with the help of FCM algorithm. Let us start with quantitative data where each observation may consists of *n* measured variables, grouped into an *n*-dimensional column vector $\mathbf{Z}_k = [z_{1k}, \ldots, z_{nk}]^T$, $\mathbf{Z}_k \in \square^n$. A set of *N* observations is denoted by $\mathbf{Z} = \{z_k \mid k = 1, 2, \ldots, N\}$, and is represented as an $n \times N$ matrix:

$$Z = \begin{bmatrix} z_{11} & z_{12} & . & . & . & z_{1N} \\ z_{21} & z_{22} & . & . & . & z_{2N} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ z_{n1} & z_{n2} & . & . & . & z_{nN} \end{bmatrix}$$

In the pattern-recognition terminology, the columns of this matrix are called patterns or objects, the rows are called the features or attributes, and **Z** is called the pattern or data matrix. The meaning of the columns and rows of **Z** depends on the context.

2. HARD AND FUZZY PARTITIONS

Hard clustering methods are based on classical set theory, and require that an object either does or does not belong to a cluster. Hard clustering means partitioning the data into a specified number of mutually exclusive subsets. Fuzzy clustering methods, however, allow the objects to belong to several clusters simultaneously, with different degrees of membership. In many situations, fuzzy clustering is more natural than hard clustering. Objects on the boundaries between several classes are not forced to fully belong to one of the classes, but rather are assigned membership degrees between 0 and 1 indicating their partial membership.

2.1. Hard Partition

The objective of clustering is to partition the data set **Z** into *c* clusters (groups, classes).Using classical sets, a hard partition of **Z** can be defined as a family of subsets $\{A_i \mid 1 \le i \le c\} \subset P(\mathbf{Z})$, (P(**Z**) is the power set of **Z**) with the following properties (Bezdek, 1981):

$$\bigcup_{i=1}^{c} A_i = Z$$

$$A_i \cap A_j = \emptyset,\ 1 \le i \ne j \le c$$

$$\emptyset \subset A_i \subset \mathbf{Z},\ 1 \le i \le c$$

(Eq: 2.1.1, 2.1.2 & 2.1.3 respectively.)

Equation (2.1.1) means that the union subsets A_i contain all the data. The subsets must be disjoint, as stated by (2.1.2), and none of them is empty nor contains all thedata in **Z** (2.1.3). In terms of membership(characteristic) functions, a partition can be conveniently represented by the partition matrix $\mathbf{U} = [\mu_{ik}]_{c\times N}$. The i_{th} row of this matrix contains values of the membership function μ_i of the i^{th} subset A_i of **Z**. It follows from the above equations that the elements of **U** must satisfy the following conditions:

$$\mu_{ik} \in \{0, 1\},\ 1 \le i \le c,\ 1 \le k \le N,$$

$$\sum_{i=1}^{c} \mu_{ik} = 1,\ \ 1 \le k \le N,$$

$$0 < \sum_{k=1}^{N} \mu_{ik} < N,\ 1 \le i \le c\ .$$

(Eq: 2.2.1, 2.2.2 & 2.2.3 respectively.)

The space of all possible hard partition matrices for **Z**, called the hard partitioning space (Bezdek, 1981), is thus defined by:

$$M_{hc} = \left\{ U \in \mathbb{R}^{c \times N} \,|\, \mu_{ik} \in \{0,1\}, \forall i, k; 0 < \sum_{k=1}^{N} \mu_{ik} < N, \forall i \right\}$$

(Eq: 2.3)

Example 1.1 Hard partition: Let us illustrate the concept of hard partition by a simpleexample. Consider a data set **Z** = {**z**1, **z**2, . . . ,**z**10}, consisting of 10 web pages each represented by graphs. Suppose we obtained the figure below after calculating the distance[2,3] between each and every pair of graphs by using the formula:

$$dist_{MCS}(z_i, z_j) = 1 - \left(\frac{\sum_{SOM} d^{\pm}(MCS(z_i, z_j))}{\max(\sum d^{\pm}(z_i), (\sum d^{\pm}(z_j)))} \right)$$

wherei,j=1,2…10

(Eq: 2.4)

as shown in Figure below:

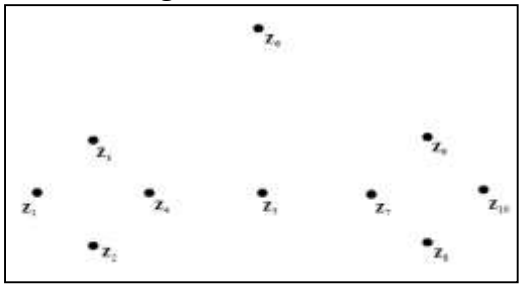

Figure 1.1. A dataset in $\mathbb{R}^2$

A visual inspection of this data may suggest two well-separated clusters (data points $\mathbf{z}_1$ to $\mathbf{z}_4$ and $\mathbf{z}_7$ to $\mathbf{z}_{10}$ respectively), one point in between the two clusters ($\mathbf{z}_5$), and an"outlier" $\mathbf{z}_6$. One particular partition $\mathbf{U} \in M_{hc}$ of the data into two subsets (out of the 2^{10} possible hard partitions) is

$$U = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

The first row of **U** defines point-wise the characteristic function for the first subset of **Z**, A_1, and the second row defines the characteristic function of the second subset of **Z**, A_2. Each sample must be assigned exclusively to one subset (cluster) of the partition. In this case, both the boundary point $\mathbf{z}_5$ and the outlier $\mathbf{z}_6$ have been assigned to A_1.It is clear that a hard partitioning may not give a realistic picture of the underlying data. Boundary data points may represent patterns with a mixture of properties of data in A_1 and A_2, and therefore cannot be fully assigned to either of these classes, or do they constitute a separate class. This shortcoming can be alleviated by using fuzzy partitions as shown in the following sections.

2.2. Fuzzy Partition

Generalization of the hard partition to the fuzzy case follows directly by allowing μ_{ik} to attain real values in [0, 1]. Conditions for a fuzzy partition matrix, analogous to (2.2) are given by (Ruspini, 1970):

$$\mu_{ik} \in [0,\ 1],\ 1 \le i \le c,\ \ 1 \le k \le N$$

$$\sum_{i=1}^{c} \mu_{ik} = 1, \underline{1} \le k \le N$$

$$0 < \sum_{i=1}^{c} \mu_{ik} = 1 < N,\ 1 \le i \le c$$

(Eq: 2.5.1, 2.5.2 & 2.5.3 respectively.)

The i_{th} row of the fuzzy partition matrix **U** contains values of the i_{th} membership function of the fuzzy subset A_i of **Z**. Equation (2.5.2) constrains the sum of each column to 1, and thus the total membership of each $\mathbf{z}_k$ in **Z** equals one. The fuzzy partitioning space for **Z** is the set

$$M_{fc} = \left\{ U \in \Box^{c \times N} \,|\, \mu_{ik} \in [0,1], \forall i,k; 0 < \sum_{k=1}^{N} \mu_{ik} < N, \forall i \right\}$$

(Eq. 2.6)

Example 1.2: Fuzzy partition: Let us consider the data set from Example 1.1. One of the infinitely many fuzzy partitions in Z is:

$$\mathbf{U} = \begin{bmatrix} 1.0 & 1.0 & 1.0 & 0.8 & 0.5 & 0.5 & 0.2 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.2 & 0.5 & 0.5 & 0.8 & 1.0 & 1.0 & 1.0 \end{bmatrix}$$

The boundary point $\mathbf{z}_5$ has now a membership degree of 0.5 in both classes, which correctly reflects its position in the middle between the two clusters. Note, however, that the outlier $\mathbf{z}_6$ has the same pair of membership degrees, even though it is further from the two clusters, and thus can be considered less typical of both A_1 and A_2 than $\mathbf{z}_5$. This is because condition (2.5.2) requires that the sum of memberships of each point equals one. It can be, of course, argued that three clusters are more appropriate in this example than two. In general, however, it is difficult to detect outliers and assign them to extra clusters.

3. FUZZY *C*-MEANS CLUSTERING

Most analytical fuzzy clustering algorithms (and also all the algorithms presented in this chapter) are based on optimization of the basic *c*-means objective function, orsome modification of it. Hence we start our discussion with presenting the FCM functional.

3.1 The Fuzzy c-Means Functional

A large family of fuzzy clustering algorithms is based on minimization of the fuzzy c-means functional formulated as (Dunn, 1974; Bezdek, 1981):

$$J(\mathbf{Z};\mathbf{U},\mathbf{V}) = \sum_{i=1}^{c} \sum_{k=1}^{N} (\mu_{ik})^m \left\| z_k - v_i \right\|_A^2$$

where

$$\mathrm{U} = [\mu_{ik}] \in M_{fc}$$

is a fuzzy partition matrix of Z,

$$\mathrm{V} = [\mathrm{v}_1, \mathrm{v}_2, \ldots, \mathrm{v}_c], \mathrm{v}_i \in \Box^n$$

is a vector of cluster prototypes(centers), which have to be determined,

$$D^2_{ikA} = \left\| z_k - v_i \right\|_A^2 = (z_k - v_i)^T A (z_k - v_i)$$

is a squared inner product distance norm where **A** is a norm-inducing matrix, and

$$\mathrm{m} \in [1, \infty)$$

(Eq: 3.1.1, 3.1.2, 3.1.3, 3.1.4 & 3.1.5 respectively.)

is a parameter which determines the fuzziness of the resulting clusters. The value of the cost function (8.1) can be seen as a measure of the total variance of z_k from v_i.

3.2. The Fuzzy c-Means Algorithm

The minimization of the *c*-means functional (3.1.1) represents a nonlinear optimization problem that can be solved by using a variety of methods, including iterative min-

imization, simulated annealing or genetic algorithms. The most popular method is a simple Picard iteration through the first-order conditions for stationary points of (3.1.1), known as the FCM algorithm.

The stationary points of the objective function (3.1.1) can be found by adjoining the constraint (2.5.2) to *J* by means of Lagrange multipliers:

$$\bar{J}(Z;U,V,\lambda) = \sum_{i=1}^{c}\sum_{k=1}^{N}(\mu_{ik})^m D^2_{ikA} + \sum_{k=1}^{N}\lambda_k\left[\sum_{i=1}^{c}\mu_{ik} - 1\right]$$

(Eq: 3.2)

and by setting the gradients of $\bar{J}$ with respect to U,V and λ to zero. It can be shown that $D^2_{ikA} > 0, \forall i,k$ and m>1, then (U,V) $\in$ $M_{fc} \times \square^{n\times c}$ may minimize if and only if

$$\mu_{ik} = \frac{1}{\sum_{j=1}^{c}(D_{ikA}/D_{jkA})^{2/(m-1)}}, 1 \le i \le c, 1 \le k \le N,$$

and $$V_i = \frac{\sum_{K=1}^{N}(\mu_{ik})^m z_k}{\sum_{k=1}^{N}(\mu_{ik})^m}; 1 \le i \le c.$$

(Eq: 3.3.1 & 3.3.2)

This solution also satisfies the remaining constraints (2.5.1) and (2.5.3). Equations (3.3)are first-order necessary conditions for stationary points of the functional (3.1.1). The FCM (Algorithm 1.1) iterates through (3.3.1) and (3.3.2). Sufficiency of (3.3) and the convergence of the FCM algorithm is proven in (Bezdek, 1980). It is to be noted that (3.3.2) gives $\mathbf{v_i}$ as the weighted mean of the data items that belong to a cluster, where the weights are the membership degrees. That is why the algorithm is called "*c*-means".

Algorithm1.1 Fuzzy*c*-means (FCM). Given the data set **Z**, choose the number of clusters $1 < c < N$, the weighting exponent $m > 1$, the termination tolerance $\in > 0$ and the norm-inducing matrix **A**. Initialize the partition matrix randomly, such that $U^{(0)} \in M_{fc}$.

Repeat for $l = 1, 2, \ldots$

Step 1: Compute the cluster prototypes (means):

$$v_i^{(l)} = \frac{\sum_{K=1}^{N}(\mu_{ik}^{(l-1)})^m z_k}{\sum_{k=1}^{N}(\mu_{ik}^{(l-1)})^m}; 1 \le i \le c.$$

Step 2: Compute the distances:

$$D^2_{ikA} = \left(z_k - v_i^{(l)}\right)^T A\left(z_k - v_i^{(l)}\right), 1 \le i \le c, 1 \le k \le N.$$

Step 3: Update the partition matrix:
for $1 \le k \le N$
if $D_{ikA} > 0$ for all $i = 1, 2, \ldots, c$

$$\mu_{ik}^{(l)} = \frac{1}{\sum_{j=1}^{c}(D_{ikA}/D_{jkA})^{2/(m-1)}},$$

otherwise,

$$\mu_{ik}^{(l)} = 0 \text{ if } D_{ikA} = 0 \text{ and } \mu_{ik}^{(l)} \in [0,1] \text{ with}$$

$$\sum_{i=1}^{c}\mu_{ik}^{(l)} = 1.$$

$$\text{until} \left\| U^{(l)} - U^{(l-1)} \right\| < \varepsilon$$

3.3. Parameters of the FCM Algorithm

Before using the FCM algorithm, the following parameters must be specified: the number of clusters, *c*, the 'fuzziness' exponent, *m*, the termination tolerance, ε, and the norm-inducing matrix, **A**. Moreover, the fuzzy partition matrix, **U**, must be initialized.

3.3.1. Number of Clusters

The number of clusters *c* is the most important parameter, in the sense that the remaining parameters have less influence on the resulting partition. When clustering real data without any a priori information about the structures in the data, one usually has to make assumptions about the number of underlying clusters. The chosen clustering algorithm then searches for *c* clusters, regardless of whether they are really present in the data or not.

3.3.2. Fuzziness Parameter

The weighting exponent *m* is a rather important parameter as well, because it significantly influences the fuzziness of the resulting partition. As *m* approaches one from above, the partition becomes hard ($\mu_{ik} \in \{0, 1\}$) and $\mathbf{v_i}$ are ordinary means of the clusters. As $m \to \infty$, the partition becomes completely fuzzy ($\mu_{ik} = 1/c$) and the cluster means are all equal to the mean of **Z**. These limit properties of (8) are independent of the optimization method used (Pal and Bezdek, 1995). Usually, $m = 2$ is initially chosen.

3.3.3. Termination Criterion

The FCM algorithm stops iterating when the norm of the difference between **U** in two successive iterations is smaller than the termination parameter ε. For the maximum norm $\max_{ik}(\left|\mu_{ik}^{l} - \mu_{ik}^{(l-1)}\right|)$, the usual choice is ε=0.001, even though $\varepsilon = 0.01$ works well in most cases, while drastically reducing the computing times.

3.3.4. Norm-Inducing Matrix

The shape of the clusters is determined by the choice of the matrix **A** in the distance measure (3.1.4). A common choice is **A** = **I**, which gives the standard Euclidean norm:

$$D_{ik}^2 = (z_k - v_i)^T (z_k - v_i)$$

3.3.5 Initial Partition Matrix

The partition matrix is usually initialized at random, such that $\mathbf{U} \in M_{fc}$. A simple approach to obtain such **U** is to initialize the cluster centers v_i at random and compute the corresponding **U** by (10.1) (i.e., by using the third step of the FCM algorithm).

4. THE MODIFIED FUZZY C MEANS ALGORITHM TO FIT WITH GRAPHS

The main challenge with adapting fuzzy c-means for graphs lies in creating a method of computing the cluster representatives.

Let us consider a graphical dataset

$Z=(z_k|k=1,2,\ldots N)$

Under fuzzy c-means the cluster centers are computed with a weighted averaging that takes into account the membership values of each data item. Thus the graph median cannot be directly used. We propose the following method of determining cluster centers for graph-based data. For each cluster *j*, use deterministic sampling to compute the number of copies of each graph *i* to use, $e_j(i)$, which is defined as:

$$e_j(i) = \left[n \frac{a_{ij}}{\sum_{\forall_i} a_{ij}} \right]$$

Here n is the total number of items in the data set. We then create a set of graphs consisting of $e_j(i)$ copies of graph i and compute the median graph of this set to be the representative of cluster j .So the new algorithm becomes:

Repeat for $l = 1, 2, \ldots$

Step 1: Compute the cluster prototypes (representative median of a set of graphs):

$$g_i^{(l)} = \arg\min_{\forall s \in S} \left(\frac{1}{|s|} \sum_{y=1}^{|S|} dist(s, G_y) \right)$$

where S is the set of graphs and $g \in S$ ($S = \{G_1, G_2, ..., G_n\}$) such that g has the lowest average distance to all elements in S[3]

Step 2: Compute the distances:

$$D^2_{ikA} = \left(z_k - g_i^{(l)}\right)^T A \left(z_k - g_i^{(l)}\right), 1 \le i \le c, 1 \le k \le N.$$

where $(z_k - g_i^l)$ is representing the distance between the graph z_k and the cluster representative g_i^l , i.e. $dist_{MCS}(z_k, g_i^l)$ (refer eq. 2.4).

Step 3: Update the partition matrix:
for $1 \le k \le N$
if $D_{ikA} > 0$ for all $i = 1, 2, \ldots, c$

$$\mu_{ik}^{(l)} = \frac{1}{\sum_{j=1}^{c} (D_{ikA} / D_{jkA})^{2/(m-1)}},$$

otherwise,

$$\mu_{ik}^{(l)} = 0 \text{ if } D_{ikA} = 0 \text{ and}$$

$$\mu_{ik}^{(l)} \in [0,1] \text{ with } \sum_{i=1}^{c} \mu_{ik}^{(l)} = 1.$$

$$\text{until} \left\| U^{(l)} - U^{(l-1)} \right\| < \varepsilon$$

4. CONCLUSION

In this article, we suggested a clustering method for graph based data with special reference to graphs representing web documents. The basic idea is the calculation of cluster center in case of graphical objects. We have modified the step 1 and 2 of the original FCM algorithm which will arm it to handle graph based data. We have made these changes without changing the fundamental concepts of the FCM algorithm. This method will enhance the efficiency and effectiveness of the FCM algorithm, as the graphical objects will boost the clustering method with abundant information [6, 7, 8].

REFERENCES

Baruah, H. K. (2011). In Search of the Root of Fuzziness: The Measure Theoretic Meaning of Partial Presence. *Annals of Fuzzy Mathematics and Informatics*, 2 (1),57- 68.

Baruah, H. K. (2011). The Theory of Fuzzy Sets: Beliefs and Realities. *International, Journal of Energy Information and Communications,* 2 (2), 1-21.

Bezdek, J. C. (1981). Pattern Recognition with Fuzzy Objective Function Algorithms. *Plenum Press.*

Cannon, R. L., Dave, J. V., Bezdek, J. C. (1986). Efficient Implementation of the Fuzzy C-Means Clustering Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8 (2), 248-255.

Dunn, J. C. (1974). A Fuzzy Relative of The Isodata Process and its Use in Detecting Compact Well Separated Clusters. *Journal of Cybernetics*, 3 (3), 32-57.

Phukon, K. K. (2012). A Composite Graph Model for Web Document and the MCS Technique. *Interna-*

tional Journal of Multimedia and Ubiquitous Engineering, 7 (1), 45-52.

Phukon, K. K. (2012). The Composite Graph Model for Web Document and its Impacts on Graph Distance Measurement. *International Journal of Energy Information and Communications*, 3 (2), 53-60.

Phukon, K. K. (2012). Maximum Common Subgraph and Median Graph Computation from Graph Representations of Web Documents Using Backtracking Search. *International Journal of Advanced Science and Technology*, 51, 67-80.

Zadeh, L. A. (1965). *Information and Control*, 338-353.