Clustering Algorithm in Data Mining Based on Web Log

Zhonghua Zhao
Department of Computer and Information Engineering, Heze University, Heze, Shandong, China
Email: 1091991039@qq.com

Abstract—The advantages of FCM algorithm are that it is mainly applied in point data cluster and can't directly process relational data, for which the paper proposes a clustering algorithm in data mining based on web log. Firstly, the paper improves FCM algorithm which makes it can process relational data, and makes robustness improvement on the algorithm. Then, the traditional FCM algorithm needs to determine in advance on the basis without prior knowledge, for which the paper introduces competition agglomerative algorithm and makes it combine with FCM algorithm, which generates CA-FCM algorithm making it can automatically determine category number of the best classification. The experiments show that mining results of CA-FCM algorithm is close to the mining results of FCM algorithm, and the performance of CA-FCM algorithm is better than that of FCM algorithm when the amount of users access to session is not too much.

Index Terms—Web Log, Data Mining, Clustering Algorithm, FCM Algorithm

I. INTRODUCTION

As the main channel for information release on the internet, Web not only has shown huge commercial value and application potential, but also the popular and important means of people acquiring information. But the change of it is huge, diversified and dynamic. With the increase of the scale and complexity of Web sits, design and maintenance of sits has become more and more difficult. Website designers try their best to optimize their own website to attract and retain more users, but it must depend on full mastery of website information. Websites operators not only need good website design, but also need to adjust web page structure dynamically according to the users’ access interest, access frequency and access time, and should improve the service for better meet the needs of visitors. And the visitors hope to use the simplest way to get the most accurate information and expect to personalization service. However, a useful tool to solve these needs is Web data mining, that is, the ideas and methods of data mining are used to mine useful information on Web.

The paper analyzes and summarizes the experience and achievements of relevant researchers, and uses fuzzy c-Means clustering algorithm to mine Web log for realizing the access to page clustering for the users. But fuzzy c-Means clustering algorithm is difficult to determine, and can't directly process relational data, but also is very sensitive to isolated points. For the above disadvantages of fuzzy c-Means clustering algorithm, we propose a improved FCM clustering algorithm based on Web log mining to find similar customers and page clustering, which provides basis for adjusting website structure and personalized service.

II. RELATED WORK

A. K-means Algorithm

Firstly, k-means algorithm randomly selects k objects, and each object represents a clustering centroid. And the other objects are distributed to the cluster which is the most similar to it according to the distance between the object and each clustering centroid. Then, the new centroid of each cluster is calculated. The above processes are repeated until the criterion functions converge. And square error criterion function is the criterion function which is usually used.

Concrete steps of k-means clustering algorithm are as follows:

1. k centroids \( C_1, C_2, \ldots, C_k \) are selected from the data set as the initial clustering center.
2. Each object is allocated to the cluster which is the most similar to it, and each syndication is represents by the mean values of all objects, and the most similar means the minimum distance. Each point \( V_i \) finds out a centroid \( C_j \), which makes the distance \( d(V_i, C_j) \) between them minimal, and \( V_i \) or allocated to the \( j \) group.
3. After all points are assigned to the corresponding groups, the centroid \( C_j \) of each group is calculated again.
4. The algorithm takes on better scalability, and the computational complexity is \( O(nkt) \) in which \( t \) is the number of times for cycle.

The disadvantage of k-means algorithm is that it not only needs to scan database for many times, and can only find out spherical classification, but also it can't find the classification with any shape. In addition, the selection of the initial centroid has great influence on clustering results. And the algorithm is very sensitive to noise.
The process of k-means algorithm is similar to that of k – means algorithm, and the unique difference between them is that k – medoids algorithm uses the object which is the most close to center to represent the cluster, but k – means algorithm uses the centroid to stand for the cluster. k-means algorithm is very sensitive to noise, the reason for which is that a great value has a great effect on the calculation of centroid. But k – medoids algorithm can effectively eliminate the influence by using center to replace centroid.

Firstly, k-means algorithm randomly selects k objects, and each object represents a clustering centroid. And the other objects are assigned to the cluster which is the most similar to it. Then each center is respectively replaced by other decentralizations to check if the quality of the cluster improves. If it improves, the replacement is reserve. The above processes are repeated until it doesn’t change any longer.

The common k – medoids algorithm includes PAM (Partitioning Around Medoids) algorithm, CLARA (Clustering LARge Application) algorithm and CLARANS (Clustering Large Application based upon RANdomized Search) algorithm. When there is noise and isolated point data, k – medoids algorithm is stronger than k – means algorithm, the reason for which is that the center points are not like mean values which are easy to be influenced by extreme data. However, the execution cost of k – medoids algorithm is higher than that of k – means algorithm.

Above all, partition method has the advantages of linear complexity and high clustering efficiency. However, as it not only demands to input the number k to determine the number of results clustering, but also it is not suitable for finding the cluster with non-convex shape or the cluster with great difference of the size, these heuristic clustering methods apply to find globular clusters in databases with small and medium size. In order to cluster the data with large scale and process the cluster with complex shape, the method based on division needs further extension.

B. FCM Fuzzy Clustering Algorithm

The paper supposes that the set of n data samples is \( X = \{x_1, x_2, \cdots, x_n\} \subset \mathbb{R}^n \), \( x_k = (x_{k1}, x_{k2}, \cdots, x_{kn})^T (\in \mathbb{R}^n) \) is the feature vector or mode vector observing sample \( x_k \), which corresponds to a point in characteristic space, \( m \) is feature dimension, \( x_{kj} \) is the assignment of feature vector \( x_k \) on the \( j \) dimension feature. \( c \) \((2 \leq c \leq n)\) is the number of category dividing the data sample, \( (P_1, P_2, \cdots, P_m)^T (\in \mathbb{R}^c, i = 1, 2, \cdots, c) \) represents the clustering prototype of the \( i \) category, and \( P = (P_1, P_2, \cdots, P_c) \).

\( (\in \mathbb{R}^{mc}) \) constructs clustering prototype matrix: \( U = (u_{ik})_{mc} \ (\in \mathbb{R}^{mc}) \) is membership matrix in which \( u_{ik} \) means the degree of membership of sample \( x_i \) for clustering prototype \( P \).

And the objective function of FCM algorithm can be represented by the following expression:

\[
\begin{align*}
\min J_m(U,P) & = \min \left\{ \sum_{i=1}^{m} \sum_{j=1}^{c} (u_{ij})^m (d_{ij})^2 \right\} \\
& = \sum_{j=1}^{c} \min \left\{ \sum_{i=1}^{m} (u_{ij})^m (d_{ij})^2 \right\}
\end{align*}
\]

(1)

In the expression, \( d_{ij} \) means the measurement of dissimilarity degree between sample \( x_i \) and the \( i \) classification of clustering prototype \( p_i \). And it is commonly Euclid distance.

\[
d_{ij} = d(x_i, p_i) = \sqrt{\sum_{j=1}^{n} (x_{ij} - p_{ij})^2}
\]

(2)

Parameter \( m \) is called smoothing factor which controls the degree of share of mode in fuzzy clustering. From the effective experimental studies of clusters, Pal [52] gets that the best selection interval of \( m \) should be and that it can take the medium value of the interval without special requirements, \( m = 2 \).

In order to make FCM algorithm and objective function achieve optimal solution, the clustering criterion can be taken: in the constraint situation of extreme value \( \sum_{i=1}^{m} u_{ik} = 1 \), making \( \min \{ J_m(U,P) \} \), that is:

\[
\min \{ J_m(U,P) \} = \min \left\{ \sum_{i=1}^{m} \sum_{j=1}^{c} (u_{ij})^m (d_{ij})^2 \right\}
\]

(3)

Therefore, the above problem can be comprehended that in the constraint situation of membership \( \sum_{i=1}^{m} u_{ik} = 1 \), evaluating:

\[
\sum_{i=1}^{m} \min \left\{ \sum_{j=1}^{c} (u_{ij})^m (d_{ij})^2 \right\}
\]

(4)

Lagrange method can be used to solve:

\[
F = \sum_{i=1}^{m} (u_{ij})^m (d_{ij})^2 + \lambda \left( \sum_{i=1}^{c} u_{ik} - 1 \right)
\]

(5)

The essential condition for optimization is:

\[
\begin{align*}
\frac{\partial F}{\partial \lambda} & = \sum_{i=1}^{m} u_{ik} - 1 = 0 \\
\frac{\partial F}{\partial u_{ik}} & = m \left( u_{ik}^{m-1} (d_{ij})^2 - \lambda \right) = 0
\end{align*}
\]

(6)

(7)
From formula (6-7), we can get \( u_p = \left[ \frac{\lambda}{m(d_p)^2} \right]^{\frac{1}{m-1}} \), which is substituted into formula (6), and we can get:

\[
\sum_{j=1}^{c} u_{ij} = \sum_{j=1}^{c} \left( \frac{\lambda}{m} \right)^{\frac{1}{m-1}} \left[ \frac{1}{(d_{ij})^2} \right]^{\frac{1}{m-1}} = \left( \frac{1}{m} \right)^{\frac{1}{m-1}} \sum_{j=1}^{c} \left[ \frac{1}{(d_{ij})^2} \right]^{\frac{1}{m-1}} = 1
\]  

(8)

So \( \left( \frac{\lambda}{m} \right)^{\frac{1}{m-1}} = \frac{1}{\sum_{j=1}^{c} \left[ \frac{1}{(d_{ij})^2} \right]^{\frac{1}{m-1}}} \) is substituted into the formula \( u_p = \left[ \frac{\lambda}{m(d_p)^2} \right]^{\frac{1}{m-1}} \), and we can achieve:

\[ u_p = \frac{1}{\sum_{j=1}^{c} \left[ \frac{1}{(d_{ij})^2} \right]^{\frac{1}{m-1}}} \]  

(9)

The similar methods can be used to achieve the value of the minimum \( p_i \) of \( J_m(U, P) \). And if \( \frac{\partial}{\partial p_i} J_m(U, P) = 0 \), we can get:

\[
\sum_{k=1}^{n} (u_{ik})^n \left( x_k - p_i \right)^T A(x_k - p_i) = 0
\]  

(10)

\[
\sum_{k=1}^{n} (u_{ik})^n = 0
\]  

(11)

And we can get the value \( u_{ik} \) of each element in membership matrix \( U \) and the clustering prototype \( P_i \):

\[ u_{ik} = \left[ \frac{\sum_{j=1}^{c} \left[ \frac{1}{(d_{ij})^2} \right]^{\frac{2}{m-1}}}{\sum_{k=1}^{n} (u_{ik})^n} \right]^{\frac{1}{2}} \]  

(12)

\[ P_i = \sum_{k=1}^{n} (u_{ik})^n x_k \]  

(13)

We could see that FCM fuzzy clustering algorithm not only has simple design and can be converted into optimization problem, but also solves with the help of nonlinear programming theory of classical mathematics and is easy to be realized for computers. Therefore, with the development and application of computers, the method becomes the hot spot of clustering research.

FCM algorithm is the clustering algorithm based on objective function optimization which is applied popularly and widely. The algorithm is developed from clustering algorithm of hard c-means. It determines the degree of each data point belonging to some cluster by membership. And the algorithm process is as follows:

Initialization: giving the category number of cluster \( c \), \( 2 \leq c \leq n \), \( n \) is the number of data. Supposing iterative stop threshold value \( \varepsilon \), initializing clustering prototype model \( P^{(0)} \) (generally \( c \) samples are selected randomly to construct \( P^{(0)} \), and the iteration counter is set as \( t = 0 \)).

Step 1: the following formula is used to calculate and update division matrix \( U^{(t)} \):

As for \( \forall i, k \), if \( \exists d_{ik}^{(t)} > 0 \),

\[ u_{ik}^{(t+1)} = \sum_{j=1}^{n} \left[ \frac{d_{ik}^{(t)}}{d_{jk}^{(t)}} \right]^{\frac{2}{m-1}} \]  

(14)

If \( \exists i, r \) make \( d_{ik}^{(t)} = 0 \), \( u_{ik}^{(t)} = 1 \) and \( j \neq r \), \( u_{ik}^{(t)} = 0 \).

Step 2: the following formula is used to update clustering prototype matrix \( P^{(t+1)} \):

\[ P_i^{(t+1)} = \frac{\sum_{k=1}^{n} (u_{ik}^{(t+1)})^n x_k}{\sum_{k=1}^{n} (u_{ik}^{(t+1)})^n} \]  

(15)

Step 3: If \( \| P^{(t)} - P^{(t+1)} \| < \varepsilon \), the algorithm stops and the division matrix \( U \) and clustering prototype \( P \) are divided. If \( t = t + 1 \), turning towards to step 2 in which \( H \) is the suitable matrix norm which is generally taken as \( F - \).

\[ \| P^{(t)} - P^{(t+1)} \| = \sqrt{\sum_{j=1}^{n} \sum_{i=1}^{n} (P_{ij} - P_{ij}^{(t+1)})^2} \]  

(16)

Finally, the value \( u_{ik} \) of each element for division matrix \( U \) and clustering prototype \( P \) are achieved:

\[ u_{ik} = \left[ \frac{\sum_{j=1}^{c} \left[ \frac{1}{(d_{ij})^2} \right]^{\frac{2}{m-1}}}{\sum_{k=1}^{n} (u_{ik})^n} \right]^{\frac{1}{2}} \]  

(17)

\[ P_i = \frac{\sum_{k=1}^{n} (u_{ik})^n x_k}{\sum_{k=1}^{n} (u_{ik})^n} \]  

(18)

FCM algorithm has the other representing method. The paper studies and improves the mining of FCM algorithm on Web log.
Step 1: Initialization. Giving the category number of cluster $c$, $2 \leq c \leq n$, $n$ is the number of data, the value of $m$ is weighted. Supposing iterative stop threshold value $\varepsilon$, initializing clustering prototype model $P^{(0)}$ (generally $c$ samples are randomly selected to construct $P^{(0)}$, and the iteration counter is set as $t$, $t=0,1,\ldots,t_{\max}$).

Step 2: Calculating the updated distance. The new clustering means the distance between data object and clustering center.

$$ P^{(i+1)}_i = \sum_{k=1}^{c} \frac{(u_{ik}^{(i)})^m}{\sum_{k=1}^{c} (u_{ik}^{(i)})^m}, 1 \leq i \leq c \quad (19) $$

$$ (d_{ik}^{(i)})^2 = \|x_k - p_{i}^{(i)}\|^2, 1 \leq i \leq c, 1 \leq k \leq n \quad (20) $$

Step 3: Updating membership matrix. The new distance is used to update clustering prototype matrix $P^{(i+1)}$.

$k=1,\ldots,n$, defines the following set:

$$ I_k = \{i \leq i \leq c : (d_{ik}^{(i)})^2 = \|x_k - p_{i}^{(i)}\|^2 = 0\} \quad (21) $$

$$ I_e = \{1,2,\ldots,c\} - I_k \quad (22) $$

If $I_k$ is null set, and

$$ u_{ik}^{(i)} = \left[\sum_{j=1}^{c} \left(\frac{(d_{ik}^{(i)})^2}{(d_{ij}^{(i)})^2}\right)^{\frac{1}{m-1}}\right]^{-1} \quad (23) $$

If $I_k$ is not null set, and

$$ \left\{\begin{array}{l}
 u_{ik}^{(i)} = 0, \forall i \in I_k \\
 \sum_{i \in I_k} u_{ik}^{(i)} = 1
\end{array}\right. \quad (24) $$

Finally, the check is made if it convergent. If $\|P^{(i)} - P^{(i+1)}\| < \varepsilon$, it is shut down, or $t=t+1$, and returns to step 2.

III. IMPROVEMENT OF FCM ALGORITHM

A. Improvement on Robustness of Algorithm

In the set of user session and user visiting pages, there may be some cases as follows: user browsing behavior represented by them is objectless navigation on internet. As for other user sessions with the purpose of access, it can be regarded as noise data.

FCM algorithm demands that membership matrix $U = [u_{ik}]_{i=1,2,\ldots,c;k=1,2,\ldots,n}$ must meet conditions $\sum_{i=1}^{c} u_{ik} = 1$, $k=1,2,\ldots,n$, and the sum of membership of each cluster attaching to noise data must be 1, which will obviously affects the accuracy of cluster, the reason for which is that the sum of membership of noise data in each cluster should be very small.

Noise clustering proposed by Dave in 1991 can be used to process noise data. And Dave proposes that noise data is included in a single noise class which makes noise data separate from other data and can't cause to reduce the quality of clustering analysis. Dave defined noise prototype as the representative of noise class, the distance from noise center to all data objects is equal, which is called noise distance). Although Dave pointed out that the noise distance can takes different values later, it is regulated as constant $\delta$ here.

$c$ is used to represent the number of good cluster which is relative to noise data, and a cluster is added, that is, the class $(c+1)$ is used to represent noise clustering, and the distance from data object $x_k$ to noise center is $d_{ik} = \delta$, the membership attaching to noise clustering is represented as $u_{ik} = 1 - \sum_{i=1}^{c} u_{ik}$, $k=1,2,\ldots,n$. And the constraint condition of membership for good cluster $\sum_{i=1}^{c} u_{ik} = 1$, $k=1,2,\ldots,n$ can be changed into $\sum_{i=1}^{c} u_{ik} \leq 1$, $k=1,2,\ldots,n$, which makes the sum of membership for noise data in good cluster arbitrarily small. And the objective functions of FCM algorithm is changed into:

$$ J_m(U,V) = \sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^m (d_{ik})^2 + \sum_{k=1}^{n} (u_{ik})^m \delta^2 \quad (25) $$

$$ (u_{ik})^m = \left(1 - \sum_{i=1}^{c} u_{ik}\right)^m, \quad \delta^2 \text{ is a constant set by the user which represents the squared distance of each data object to noise center.} $$

Its significance is that if the squared distance of a data object to any good cluster is greater than $\delta^2$, the data objects is seen noise data, and the degree attaching to noise clustering is the biggest.

FCM algorithm needs to change as follows:

In step 1, the value of noise clustering $c$ and noise distance $\delta$ are determined, and the initialization membership matrix is $U = [u_{ik}]_{i=1,2,\ldots,c;k=1,2,\ldots,n}$.

In step 2,

$$ d_{ik}^{(i+1)} = \delta \quad (26) $$

And $k=1,2,\ldots,n$.

In step 3, $c+1$ replaces $c$:

$$ u_{ik}^{(i)} = \left[\sum_{j=1}^{c} \left(\frac{1}{(d_{ik}^{(i)})^2}\right)^{\frac{1}{m-1}}\right]^{-1} \quad (27) $$

$$ \sum_{j=1}^{c} \left(\frac{1}{(d_{ik}^{(i)})^2}\right)^{\frac{1}{m-1}} + \left(\frac{1}{\delta^2}\right)^{\frac{1}{m-1}} \quad (26) $$
And the membership of data objects attaching to noise clustering can be calculated:

\[
U_{i k}^{(t)} = \frac{\left[ \frac{1}{\delta^2} \right]^{\frac{1}{(m-1)}}}{\sum_{j=1}^{\infty} \left[ \frac{1}{(d^{(t+1)}_{i j})^2} \right]^{\frac{1}{(m-1)}} + \left[ \frac{1}{\delta^2} \right]^{\frac{1}{(m-1)}}}
\]  
(28)

The selection of \( \delta \) is a very complicated problem, which needs to estimate the size of each cluster. However, in the actual problem, noise distance can take the follow constant:

\[
\delta^2 = \lambda \left[ \sum_{i=1}^{n} \sum_{k=1}^{m} (d_{ik})^2 \right] / cn
\]  
(29)

\( \lambda \) is proportionality coefficient which needs to be selected according to the type of the data to be clustered, and it reduces gradually with the algorithm.

It is more direct for noise clustering method to apply to objective data compared with relational data, the reason for which is that there is clustering center in relational data cluster in the strict sense, so there is no noise center. In relational cluster, Noise clustering is defined as the cluster with the same dissimilarity degree between any two data objects, that is, * is used to replace noise clustering, \( (R_{\beta}) = \delta (1 \leq j \leq n, 1 \leq k \leq n) \), in which \( \delta \) is constant and is called dissimilarity degree of noise. According to the definition, the dissimilarity degree between any two data objects in noise clustering is \( \delta \). Therefore, the dissimilarity degree between any two data objects in a good cluster can't be greater than \( \delta \).

Obviously, if the dissimilarity degree between the data objects of each good cluster and some data object of the data concentration to be clustered is greater than \( \delta \), the data object should be divided into noise clustering. On the other hand, if the dissimilarity degree between some data object and at least one data object in a good cluster is less than \( \delta \), the data object should belong to the good cluster and not belong to noise clustering.

Noise clustering (NC) is applied in FCM algorithm of relational data, and the objective function should be changed into:

\[
J_m(U,V) = \sum_{i=1}^{n} \sum_{k=1}^{m} (u_{ik})^m (d_{ik})^2 + \sum_{k=1}^{m} (u_{ik})^m (d_{ik})^2
\]  
(30)

In the formula, * means noise clustering. Hathaway deduced \( d_{ik}^2 = \frac{1}{2} \delta \), so FCM algorithm of relational data needs to do the following changes:

In step (1), the value of noise clustering \( c \) and noise distance \( \delta \) are determined, and the initialization membership matrix is \( U = [u_{ik}]_{1 \leq i \leq n , 1 \leq j \leq m} \).

In step 2, \( d^{(t+1)}_{i k} = \delta \) and \( k = 1, 2, \ldots, n \).

In step 3, the following formula replaces, and the membership of data object attaching to good cluster is calculated as follows:

\[
U_{i k}^{(t)} = \frac{\left[ \frac{1}{\delta^2} \right]^{\frac{1}{(m-1)}}}{\sum_{j=1}^{\infty} \left[ \frac{1}{(d^{(t+1)}_{i j})^2} \right]^{\frac{1}{(m-1)}} + \left[ \frac{2}{\delta^2} \right]^{\frac{1}{(m-1)}}}
\]  
(31)

And the membership of data object attaching to noise clustering can be calculated:

\[
U_{i k}^{(t)} = \frac{\left[ \frac{2}{\delta^2} \right]^{\frac{1}{(m-1)}}}{\sum_{j=1}^{\infty} \left[ \frac{1}{(d^{(t+1)}_{i j})^2} \right]^{\frac{1}{(m-1)}} + \left[ \frac{2}{\delta^2} \right]^{\frac{1}{(m-1)}}}
\]  
(32)

The above changes and the introduction of a new noise clustering can form FCM algorithm of strong relational data which can process the data set including noise. And the paper calls it as strong FCM algorithm of relational data.

B. Realization of CA-FCM Algorithm

In the CA-FCM algorithm of robust relational data, the objective function is added a feature, which makes it as:

\[
J = \sum_{i=1}^{n} \sum_{k=1}^{m} (u_{ik})^m (d_{ik})^2 - \alpha \sum_{k=1}^{m} \sum_{i=1}^{n} (u_{ik})^m + \left( \sum_{i=1}^{n} u_{ik} \right)^2
\]  
(33)

as a new objective function, the iterative method is used to evaluate the approximate value of the minimum. And the steps of algorithm are as follows:

(1) Initialization. \( m = 2 \), and the objective function is:

\[
J = \sum_{i=1}^{n} \sum_{k=1}^{m} (u_{ik})^m (d_{ik})^2 - \alpha \sum_{k=1}^{m} \sum_{i=1}^{n} (u_{ik})^m + \left( \sum_{i=1}^{n} u_{ik} \right)^2
\]  
(34)

In the formula, * represents noise clustering, the iteration counter \( t \) is 0, and \( c \) value, \( \eta_0 \) value, \( \tau \) value and \( \lambda \) value of the maximum clustering category number are determined, the membership matrix \( P^{(0)} \) is initialized. The minimal threshold of class base is \( n_z \), iterative stop threshold value is \( \varepsilon \). \( \beta = 0 \), and \( R_{\beta} \) is the original dissimilarity matrix \( R = [R_{\beta}]_{n \times n} \). \( P^{(t)} \) need to meet the following formulas:

\[
\sum_{i=1}^{n} u_{ik} + u_{ik} = 1, \quad k = 1, 2, \ldots, n
\]  
(35)
0 ≤ u_{ik} ≤ 1, \  i = 1, 2, \ldots, c, \ k = 1, 2, \ldots, n
0 ≤ u_{ik} ≤ 1, \  k = 1, 2, \ldots, n

Calculating the cardinality of the cluster  \( i \) :
\[
 n_{i}^{(i+1)} = \sum_{k=1}^{n} u_{ik}^{(i+1)}, \ i = 1, 2, \ldots, c
\]  (36)

Calculating the cardinality of noise cluster:
\[
 n_{n}^{(i+1)} = \sum_{k=1}^{n} u_{ik}^{(i+1)}
\]

Updating the distance, and calculating the new membership vector and the distance between data objects and cluster:
\[
 p_{i}^{(i+1)} = \left( \sum_{k=1}^{n} u_{ik}^{(i+1)} \right)^{2}, 1 \leq i \leq c
\]  (37)
\[
 (d_{ik}^{(i+1)})^{2} = (R_{ik}^{(i+1)})^{2} - \frac{1}{2} \left( \beta p_{i}^{(i+1)} \right) + \frac{\Delta \beta}{2} \left( p_{i}^{(i+1)} - \epsilon_{k} \right)
\]  (38)

As for any \( i \) and \( k \), if \( (d_{ik}^{(i+1)})^{2} < 0 \), calculating:
\[
 \Delta \beta = \max \left\{ -2(d_{ik}^{(i+1)})^{2} \right\} \left\| p_{i}^{(i+1)} - \epsilon_{k} \right\|
\]  (39)
\[
 (d_{ik}^{(i+1)})^{2} = (\beta_{ik}^{(i+1)})^{2} + \frac{\Delta \beta}{2} \left( p_{i}^{(i+1)} - \epsilon_{k} \right)
\]  (40)
\[
 \beta = \beta + \Delta \beta
\]  (41)
\[
 R_{ik}^{(i+1)} = \begin{cases} R_{ik}^{(i+1)} + \beta, & j \neq k \\ 0, & j = k 
\end{cases}
\]  (42)

\( \epsilon_{k} \) in the above formula represents the vector of the \( k \) column of unit matrix in \( R^{c} \).

Determining the distance \( \delta \) from data object to noise clustering:
\[
 \delta^{2} = \lambda \left[ \sum_{i=1}^{c} \sum_{k=1}^{n} (d_{ik}^{(i+1)})^{2} \right]
\]  (43)
\[
 (d_{ik}^{(i+1)})^{2} = \frac{1}{2} \delta
\]  (44)

Updating \( a(t+1) \) and membership matrix:
\[
 n_{i}^{(i+1)} = \eta_{ik} e^{-a(t+1)/\tau}
\]  (45)
\[
 a(t+1) = \frac{\sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^{2} (d_{ik}^{(i+1)})^{2} + \sum_{i=1}^{c} (u_{ik})^{2} (d_{ik}^{(i+1)})^{2}}{\sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^{2} + \sum_{i=1}^{c} (u_{ik})^{2}}
\]  (46)

\( k = 1, 2, \ldots, n \) define the following set:
\[
 I_{k} = \{ 1 \leq i \leq c : (d_{ik}^{(i+1)})^{2} = 0 \}
\]  (47)
\[
 \overline{I}_{k} = \{ 1, 2, \ldots, c \} - I_{k}
\]  (48)

If \( u_{ik}^{(i)} > 1 \Rightarrow u_{ik}^{(i)} = 1 \)
and \( u_{ik}^{(i)} < 0 \Rightarrow u_{ik}^{(i)} = 0 \).
\[
 u_{ik}^{(i)} = \frac{1}{\delta} \left[ \sum_{j=1}^{c} (d_{jk}^{(i+1)})^{2} + \frac{1}{\delta} + \frac{a(t+1)}{\delta} \right] (n_{j}^{(i)} - n_{j}^{(i+1)})
\]

If \( u_{ik}^{(i)} > 1 \Rightarrow u_{ik}^{(i)} = 1 \), and \( u_{ik}^{(i)} < 0 \Rightarrow u_{ik}^{(i)} = 0 \).
\[
 I_{k} \text{ is not null set} \Rightarrow \sum_{i=1}^{c} \eta_{ik}^{(i)} = 1
\]

Updating clustering number and calculating clustering cardinality:
\[
 n_{i}^{(i)} = \sum_{k=1}^{c} \eta_{ik}^{(i)}, \ i = 1, 2, \ldots, c
\]  (49)

Calculating the cardinality of noise clustering:
\[
 n_{n}^{(i)} = \sum_{k=1}^{n} u_{ik}^{(i)}
\]  (50)

If \( n_{i}^{(i)} < n_{i} \) and \( i = 1, 2, \ldots, c \), the cluster \( i \) is discarded and the clustering category number \( c \) is updated.
It is checked to find if it is convergent. If the clustering category number \( c \) doesn't change and \( \| p^{(i)} - p^{(i+1)} \| < \epsilon \), the calculation stops, or \( t = t+1 \) and returning to the second step.

<table>
<thead>
<tr>
<th>TABLE I. SIMILARITY MATRIX OF USERS VISITING SESSIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session</td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>S1</td>
</tr>
<tr>
<td>S2</td>
</tr>
<tr>
<td>S3</td>
</tr>
<tr>
<td>S4</td>
</tr>
<tr>
<td>S5</td>
</tr>
<tr>
<td>S6</td>
</tr>
<tr>
<td>S7</td>
</tr>
<tr>
<td>S8</td>
</tr>
</tbody>
</table>

IV. SIMULATION RESULTS

The paper uses Visual C++ 6.0 to realize the above-mentioned similarity algorithm of user access, and uses some data of data table session_20071215 as
experimental data to get the similarity of user sessions. The results are represented by similarity matrix to be as the input of user session clustering algorithm.

Because of the limitation of the space, Table 1 only gives partial similarity matrix. S1, S2, ..., S8 in the table are the number of user sessions. Because the similarity matrix is symmetric matrix, the table only gives the upper triangular matrix.

According to the membership matrix of the fourth iteration, each user visiting session is assigned to the cluster with the maximum membership, and 7 clusters are achieved, as shown in Table 2.

**TABLE II. SUMMARY OF CLUSTERING RESULTS**

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Sessions belonging to the cluster</th>
<th>Number of sessions</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>S4, S6, S11, S12, S26, S35</td>
<td>6</td>
</tr>
<tr>
<td>C2</td>
<td>S1, S3, S7, S14, S16, S17, S25, S36, S39, S43, S44, S46, S49</td>
<td>13</td>
</tr>
<tr>
<td>C3</td>
<td>S1, S9, S10, S21, S27, S31, S33, S41, S50</td>
<td>9</td>
</tr>
<tr>
<td>C4</td>
<td>S3, S15, S38</td>
<td>3</td>
</tr>
<tr>
<td>C5</td>
<td>S1, S5, S6, S7, S13, S18, S19, S30, S34, S40, S43, S45, S47, S51, S52</td>
<td>15</td>
</tr>
<tr>
<td>C6</td>
<td>S5, S22</td>
<td>2</td>
</tr>
<tr>
<td>C7</td>
<td>S2, S8</td>
<td>10</td>
</tr>
</tbody>
</table>

In the above table, most weights of URL in the cluster C4 is less than 0.2, which can be judged as noise cluster. Respectively the intra-class similarity of each cluster $S_{int}$ and similarity between classes $S_{inter}$ are figured out, as shown in Table 3.

**TABLE III. INTRA-CLASS SIMILARITY AND SIMILARITY BETWEEN CLASSES FOR EACH CLUSTER**

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Intra-class similarity</th>
<th>Similarity between classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>0.427</td>
<td>0.079</td>
</tr>
<tr>
<td>C2</td>
<td>0.213</td>
<td>0.061</td>
</tr>
<tr>
<td>C3</td>
<td>0.335</td>
<td>0.032</td>
</tr>
<tr>
<td>C4</td>
<td>0.056</td>
<td>0.042</td>
</tr>
<tr>
<td>C5</td>
<td>0.204</td>
<td>0.093</td>
</tr>
<tr>
<td>C6</td>
<td>0.372</td>
<td>0.041</td>
</tr>
<tr>
<td>C7</td>
<td>0.162</td>
<td>0.084</td>
</tr>
<tr>
<td>Mean</td>
<td>0.253</td>
<td>0.072</td>
</tr>
</tbody>
</table>

From the calculation results of intra-class similarity in the above table, the intra-class similarity of cluster C4 is evidently lower than that of other clusters, so it is noise cluster, which is the same as the result of Table 2. The mean value of intra-class similarity and similarity between classes are the mean value after removing the noise cluster. Although the absolute quantity of the mean value (0.253) of intra-class similarity is not larger, but it is larger compared with the average similarity of all sessions between two users. The mean value of intra-class similarity for each clustering is more close to the average similarity of sessions between two users. And we can know that the clustering result is better.

The time of collecting data in the experiment was 16:30:00—18:30:00 December 15th 2007 to December 20th 2007 in which most students ate dinner. As the students were in the stage of final examination, the students mainly visited undergraduate teaching, graduate information network for inquiring information relation to final examination, and their department of mathematics.

Therefore, mining results of FCM algorithm can be understood and is an accurate reflection of the mode of all user accessing. The clustering results are near to that of CA-WFCM algorithm proposed in the paper, which proves the feasibility and correctness of CA-WFCM algorithm.

In order to compare operation performance of CA-WFCM and FCM algorithm, the paper respectively selected 100, 300, 500, 1000, 2000, 5000, 7000, 10000 processed web log user sessions to compare the cluster running time. And the running time of using CA-WFCM and FCM algorithm was respectively recorded. The results are shown in the following figure 1.

**Figure 1. Comparison of CPU execution time between algorithms**

From the above figure, we can see that when the amount of log data is less than 5000, the rising rend of CPU execution time curve for CA-WFCM algorithm is slow, and the performance is better that traditional FCM algorithm. But when the amount of log data is greater than 5000, the rising rend of CPU execution time curve for CA-WFCM algorithm is rapid, and the performance is worse than FCM algorithm. The reason is that the update steps in the process of iteration of CA-WFCM algorithm and FCM algorithm are the same, and CA-WFCM can use competitive agglomeration mechanism to automatically determine the best number of clustering categories without needing priori knowledge like FCM algorithm, but time complexity of CA-WFCM algorithm calculating user sessions is $O(n^2)$ which is higher than FCM algorithm. It indicates that CA-WFCM algorithm needs further improvement in executing high-volume Web log data mining, which is the future key research content of the paper.

V. CONCLUSION

The paper improves FCM algorithm based on the characteristics of Web log data, which makes it can directly cluster the relational data. And for the disadvantage of FCM algorithm that it is difficult to determine in advance the clustering category number $c$, the paper uses the mechanism based on competition condensation algorithm combined with FCM algorithm, which forms CA-FCM algorithm and makes robust improvement on algorithm. At last, the paper makes detailed experiments and analysis on the algorithm, and
compares it with the results of FCM algorithm, which proves the feasibility and correctness of the algorithm.

Based on detailed analysis of the advantages and disadvantages of the existing clustering algorithms, the paper selected FCM algorithm to study and improve Web log data according to its characteristics. FCM algorithm is mainly applied in data cluster and can’t directly process relational data, for which the paper firstly improves FCM algorithm to make it can process relational data. Traditional FCM algorithm needs to predetermine the number of clustering categories without prior knowledge, for which the paper introduces CA algorithm. The algorithm is combined with FCA algorithm and CA-FCM algorithm generates, which makes it can automatically determine the number of the best category. And the paper adds a weight to membership to reduce the influence of outlier data on aggregative center. Finally, the paper proposes weighted CA-WFCM algorithm based on CA for the characteristics of Web data log.

REFERENCES