Fuzzy *K*-Means Incremental Clustering Based on *K*-Center and Vector Quantization

Taoying Li and Yan Chen
Transportation Management College, Dalian Maritime University, Dalian 116026, P.R. China
ytaoli@126.com

***Abstract*—Fuzzy *k*-means and vector quantization are combined in this paper to complement each other in incremental mode because each has qualities which the other lacks. The threshold of vector quantization is given and the pattern of computing the distance between the new coming data point and the k centers is introduced in a new way. We firstly reduce redundant attributes and eliminate the difference of units of dimensions and make units of all attributes same. Then, we use *k*-center to produce initial *k* means and partition data points into no more than *k* clusters. Besides, we adopt vector quantization to classify incremental data points and then adjust means after the structure of clustering varying. Finally, it is applied to real datasets and results show its efficiency and precision.**

***Index Terms*—data mining; fuzzy clustering; *k*-means algorithm; incremental clustering; vector quantization.**

I. INTRODUCTION

Clustering plays an important role in data mining and is applied widely in fields of pattern recognition, computer version, and fuzzy control. Clustering is to divide data points into groups of data points and pursues the intra-cluster similarity minimum and the cross-cluster similarity maximum [1-3]. Various types of clustering methods have been proposed and developed [4]. Clustering algorithms are mainly divided into five categories that are hierarchical, partitioning, density-based, grid-based and model-based clustering. Hierarchical and partitioning clustering methods are commonly used in practice. Hierarchical algorithm can be further divided into bottom-up and top-down algorithms [1]. Traditional hierarchy clustering algorithms are not suitable to large dataset because of too computationally intensive, such as BIRCH [2] and CURE [3]. CLIQUE [4], ENCLUS, and MAFIA [5] belong to bottom-up algorithms. PROCLUS [6] and ORCLUS [7] belong to top-down algorithms. Traditional partition clustering algorithms are *k*-means, *k*-modes, and so on. *K*-means is the most classical one among existing algorithms.

With the development of information technology, especially with the appearance of Web, data and environment are varying from minute to minute and more and more space is needed for storing data in memory. Then, incremental clustering was proposed because of the advantage of limited space requirement since the entire dataset is not necessary to store in the memory [8].

Incremental clustering has attracted a substantial amount of attention since Hartigan's algorithm [9] was implemented in [10]. D. Fisher proposed COBWEB [11], an incremental clustering algorithm that involved restructurings of the clusters in addition to the incremental additions of objects. Incremental clustering related to dynamic aspects of databases were discussed in [12-13] and was widely used in many fields [14-17]. The drive force for interest in incremental clustering is that the main memory usage is minimal since there is no need to keep in memory the mutual distances between objects and the algorithms are scalable with respect to the size of the set of objects and the number of attributes [18].

The research on incremental clustering focuses on taking the incremental data into time serial data or under special sequence and it is presently divided into two groups. One is to partition all data points iteratively, starting with the first data point to the last point, the advantage of which is high precision, but it doesn't make use of the results of last clustering and waste resources. Another is to make use of the results of last clustering and consider input one at a time and assign it to the existing clusters [19], which means that new coming data points are partitioned into the clusters whose centers are closest to them. Then we move these centers to the new coming data points, which means that a new input data points is assigned to a cluster without affecting the existing clusters significantly and just keep the structure of the clustering change little even if the new coming data points are totally different. According to the analysis mentioned above, both of clustering algorithms mentioned above have drawbacks.

In this paper, we propose a fuzzy *k*-means incremental clustering based on k-center and vector quantization, which connects both of methods, mentioned above, and overcomes their shortcomings. We start with removing redundant attributes and eliminating difference of units of dimensions. Then we use fuzzy *k*-means algorithm to group existing data points into no more than *k* clusters. Finally, we partition incremental data points into its clusters and adjust their structure.

The reminder of this paper is arranged as follows. In section 2, we describe a brief review of fuzzy k-means clustering algorithm and incremental clustering, and analyze their problems. Then the fuzzy k-means algorithm and incremental clustering are presented in section 3. In section 4, we apply the proposed clustering

doi:10.4304/jcp.5.11.1670-1677

algorithm to partition Iris dataset, Pima-Indians-Diabetes dataset and Segmentation dataset. In section 5, we give the conclusion according to section 4.

II. Related Work

In this section, we firstly review research on k-means algorithm and its variations, and then present the general process of incremental clustering.

A. Fuzzy K-Means Algorithms

The *k*-means algorithms, like other partition clustering algorithms, group the data points into *k* clusters by minimizing a cost function that has been pre-designed. The type of traditional cost function [20] is like (1).

$$F_1(C) = \sum_{j=1}^{n}\sum_{i=1}^{m}(x_{ji} - c_{li})^2 \quad (1)$$

where x_{ji} is the value of the *i*th dimension of the *j*th object. c_l is the center that nearest to the *j*th object and c_{li} is the value of the *i*th dimension of the *l*th cluster center. Because the significances of different dimensions contributing to the clustering and the preferences of each object belonging to a cluster are different, the extension with weights of the traditional cost function is often used. The methods used in [21-25] are all the extensions of (1). H. Friguiand and O. Nasraoui [22], Y. Chan and W. Ching [23] introduce the degree of membership for each object belonging to every cluster and the weight for each dimension of a cluster on contributing to clustering. However, their algorithm is not computable if one of weight happens to be zero. Domeniconi [24-25] introduces a cost function with maximum function and was proved difficult to solving the minimum objective function. [1] introduces a cost function avoiding the problems above, and they use entropy of the dimension weights to represent the certainty of dimensions in the identification of a cluster. However, the goal of clustering is to make the distance of objects in a cluster as small as possible while make the distance of objects between different clusters as large as possible [26]. The cost function in [1] does not satisfy the second part. We proposed a new method by adding a variable to adjust its function [27], which satisfies the goal of the clustering and was proved effective by some different datasets.

B. Incremental Clustering

Essentially, the incremental clustering is problem of maintaining or changing the structure of *k* clusters. For example, a new point in a particular sequence may be assigned to one of the existing *k* clusters or a new cluster or at least two other existing clusters are collapsed into one [28]. Algorithms for solving problem of incremental clustering have been studied in [16, 28].

Incremental clustering was firstly advanced in [17], and it was incremental DBSCAN based on DBSCAN. Due to the density-based nature of DBSCAN, the insertion or deletion of a data point affects the current clustering only in the neighborhood of this point, which is high accuracy because its results are similar to that of non-incremental clustering, however, it can only be used to partition data one by one whose efficiency is very low. Reference [29] proposed an incremental clustering based on grid, which was similar to incremental clustering. Huang and Zou, and Xu and Xie [30-31] adopted incremental clustering based on density in a batch mode, which could process data points in a batch, not one by one. However, they were too computationally intensive to partition large dataset. An incremental algorithm of high efficiency for clustering based on density was described in [32], which made use of partitioning and sampling technology to process large dataset and produced sampling error while partitioning high dimensions. Hsu and Huang [8] used conceptual hierarchy tree to solve similar degrees of data mixed numeric attributes and categorical attributes, but weights are required to give in advance, which means that users must have the knowledge of a particular area clearly and limits the application of the clustering.

We propose a *k*-means algorithm for incremental clustering, which makes use of *k*-means algorithm to partition history data iteratively in database, and then adjusts the structure of clustering using new coming data points by merging, deleting and adding cluster to the existing results.

III. Fuzzy K-Means Incremental Clustering Based on K-Center and Vector Quantization

A. Fuzzy K-Means

Because of the difference of the contribution of different dimensions to the clustering and the preferences of each object belonging to a cluster, the cost function (2) is proposed by [22-23]. In their research, the clustering is best while the objective function F_2 is the minimum.

$$F_2(T,W,C) = \sum_{l=1}^{k}\sum_{j=1}^{n}\sum_{i=1}^{m}\tau_{lj}\omega_{li}^{\beta}(x_{ji} - c_{li})^2 \quad (2)$$

Subject to

$$\sum_{l=1}^{k}\tau_{lj} = 1, 1 \le j \le n, \tau_{lj} \in \{0,1\}, \sum_{i=1}^{m}\omega_{li} = 1, 0 \le \omega_{li} \le 1, 1 \le l \le k$$

Where k , n and m are respectively the number of clusters, objects, and dimensions. x_{ji} is the value of the *i*th dimension of the *j*th object. C=[c_{li}] is a *k*-by-*m* matrix, and c_{li} is the value of the *i*th dimension of the *l*th cluster center. T=[τ_{lj}] is a *k*-by-*n* matrix, and τ_{lj} is the degree of membership of the *j*th object belonging to the *l*th cluster. W=[ω_{li}] is a *k*-by-*m* matrix, and ω_{li} is the weight of the *i*th dimension in the *l*th cluster. β is a parameter that greater than 1.

In (2), T, W and C are unknowns, and each of them can be solved by fixed others, which can be shown as following:

$$\tau_{lj} = \begin{cases} 0 & x_j \notin C_l \\ 1 & x_j \in C_l \end{cases}, \; c_{li} = \frac{1}{\sum_{j=1}^{n}\tau_{lj}}\sum_{j=1}^{n}\tau_{lj}x_{ji} \quad (3)$$

$$\omega_{li}=\sum_{t=1}^{m}\left[\frac{\sum_{j=1}^{n}\tau_{lj}(c_{li}-x_{ji})^2}{\sum_{j=1}^{n}\tau_{lj}(c_{lt}-x_{ji})^2}\right]^{1/(\beta-1)}$$

As analyzing in [1], ω_{li} is not computable if the dispersion of a dimension in a cluster happens to be zero because of the zero denominators.

The Locally Adaptive Clustering (LAC) algorithm for a minimization problem is proposed by Domeniconi in her dissertation [24] and other researches [25]. The cost function of the LAC algorithm is shown as following:

$$F_3(C,W)=\sum_{l=1}^{k}\sum_{i=1}^{m}\omega_{li}e^{y_{li}} \tag{4}$$

Where

$$y_{li}=\frac{1}{n_l}\max_{i'}\{\sum_{c(j)=l}(c_{li'}-x_{ji'})^2\}-\frac{1}{n_l}\sum_{c(j)=l}(c_{li}-x_{li})^2 \tag{5}$$

Subject to $\sum_{i=1}^{m}\omega_{li}^2=1, 0\le\omega_{li}\le 1, 1\le l\le k$.

Here, $c(j)=l$ means that the jth object is assigned to the lth cluster, otherwise, it is not assigned to the lth cluster. n_l is the number of objects in the lth cluster.

The LAC algorithm overcomes the problem that the denominators may be zeros. However, the presentation of $F_3(C, W)$ is not integral and differential because of the maximum function.

Liping Jing, Michael K. Ng, and Joshua Zhexue Huang proposed the entropy weighting k-means (EWKM) algorithm in their paper [1] by adding the weight entropy to modify the cost function (2) in [1], and the modified cost function can be shown as (6).

$$F_3(T,W,C)=\sum_{l=1}^{k}\left[\sum_{j=1}^{n}\sum_{i=1}^{m}\tau_{lj}\omega_{li}(c_{li}-x_{ji})^2+\gamma\sum_{i=1}^{m}\omega_{li}\log\omega_{li}\right] \tag{6}$$

Subject to $\sum_{l=1}^{k}\tau_{lj}=1, 1\le j\le n, \tau_{lj}\in\{0,1\}, \sum_{i=1}^{m}\omega_{li}=1, 0\le\omega_{li}\le 1, 1\le l\le k$.

Here, the strength of the incentive for clustering on more dimensions is controlled by the condition of the parameter $\gamma>0$.

In EWKM algorithm, the weights of dimensions represent their contribution for forming the cluster and their entropy stand for the certainty of dimensions in the identifying a cluster.

All these clustering algorithms satisfy the former part of the goal that the distance between any two objects in a cluster is as small as possible. However, making the distance of cross-cluster as large as possible is not satisfied.

For the reasons mentioned above, we introduced the improved entropy weighting k-means algorithm in [27]. In the new algorithm, we extended EWKM algorithm by modifying the cost function F_3 with an adjusted part.

The new cost function can be given as follows:

$$F(T,W,C)=\sum_{l=1}^{k}\left[\frac{\sum_{j=1}^{n}\sum_{i=1}^{m}\tau_{lj}\omega_{li}(c_{li}-x_{ji})^2}{\sum_{i=1}^{m}(c_{li}-\bar{x}_i)^2}\right]+\sum_{l=1}^{k}[\gamma\sum_{i=1}^{m}\omega_{li}\log\omega_{li}] \tag{7}$$

Subject to $\sum_{l=1}^{k}\tau_{lj}=1, 1\le j\le n, \tau_{lj}\in\{0,1\}, \sum_{i=1}^{m}\omega_{li}=1, 0\le\omega_{li}\le 1, 1\le l\le k$.

Here, $\bar{x}$ is the mean of all objects, $\bar{x}_i$ is value of the ith dimension of $\bar{x}$ and it equals to $\bar{x}_i=\frac{1}{n}\sum_{j=1}^{n}x_{ji}$. If $n>1$, the new algorithm is available, otherwise, $\sum_{i=1}^{m}(c_{li}-\bar{x}_i)^2$ is zero and the cost function $F(T,W,C)$ is not computable. The denominator is a variable and is linear to the square sum of the distances from the mean of all objects to the means of all clusters.

Next, we use the improved entropy weighting k-means algorithm to solve the minimization problem.

Minimization of F in (7) with constraints forms a class of constrained nonlinear optimization problems whose solutions are unknown. Generally, the method for solving optimization is partial optimization for T, W and C. Methods in [1], [22] can be used for reference. Thus we first fix T and C, and search appropriate W to minimize $F(T,W,C)$. Then we fix T and W, and search appropriate C. Later, we fix C and W and get appropriate T.

We repeat until the value of objective function can't decrease any more.

Theorem1.

Let T and C be fixed, F is the minimum if the weight

$$\omega_{lt}=\frac{1}{\sum_{i=1}^{m}\exp(\frac{-\psi_{li}}{\gamma})}\exp(\frac{-\psi_{lt}}{\gamma}) \tag{8}$$

here,

$$\psi_{lt}=\frac{1}{\sum_{i=1}^{m}(c_{li}-\bar{x}_i)^2}\sum_{j=1}^{n}\tau_{lj}(c_{lt}-x_{ji})^2 \tag{9}$$

We use Lagrange multiplier technique to gain the minimization problem without constraints, and its process of proof is similar to process in [1], [23].

And we know that the jth object belongs to the lth cluster if the distance between the jth object and the mean of the lth cluster is smallest, which equals to (10).

$$\tau_{lj}=\begin{cases}1, & if\ \sum_{i=1}^{m}\omega_{li}(c_{li}-x_{ji})^2\le\sum_{i=1}^{m}\omega_{zi}(c_{zi}-x_{ji})^2\\ 0, & otherwise\end{cases} \tag{10}$$

$\tau_{lj}=1$ means that the jth object belongs to the lth cluster, or not belongs to the lth cluster.

Let T and W be fixed, we use the method of solving average in mathematic to obtain the value of C as shown in (11).

$$c_{li}=\frac{1}{\sum_{j=1}^{n}\tau_{lj}}\sum_{j=1}^{n}\tau_{lj}x_{ji} \tag{11}$$

Here, $1\le l\le k$ and $1\le i\le m$.

We presented the process of the weighted fuzzy k-means algorithm in [27] as follows:

Step1. Input the parameters *m*, *n*, *k*, γ and the max iterative time *s*, init initial weights $\omega_{li}=1/m$, and choose *k* objects randomly as the centers *C* of *k* clusters.

Step2. Obtain *T* according to (10);

Step3. Compute the value of *F*(*T*,*W*,*C*) according to (7);

Step4. Updates *C* according to (11);

Step5. Update *W* according to (10);

Step6. Repeat Step2 to Step5 until the *F*(*T*,*W*,*C*) can't decrease or the iterative time beyond S.

In order to reduce the time of iteration of weighted fuzzy *k*-means, we made some adjustments by using *k*-center algorithm to choose initial *k* points, and the process of weighted *k*-means algorithm can be shown as follows:

Step1. Input the parameters *m*, *n*, *k*, γ and the max iterative time *s*, init initial weights $\omega_{li}=1/m$.

Step2. We put all data points into the set of *H* and let set of centers *C* be empty. Then, we choose one point randomly from *H* as the first center, and put it in the set of centers *C* at the same time remove it from *H*.

Step3: Make the data point from *H* which is farthest to the center data points in *C* be the next center and put it in *C* and remove it from *H*.

Step4: If the number of centers in *C* equals to *k*, go to Step5, else got o Step3.

Step5 Obtain *T* according to (17); Compute the value of *F*(*T*,*W*,*C*) according to (7);

Step6. Updates *C* according to (18); Update *W* according to (8);

Step7. Repeat Step5 to Step6 until the *F*(*T*,*W*,*C*) can't be improved or the iterative time is greater than S.

The complexity of traditional clustering based on distance is O (mn^2), and it changes exponentially along with the number of objects needing to be partitioned. Thereby, traditional clustering methods need plenty of calculation while a lot of objects exist. The complexity of the proposed algorithm equals to O (*mnk*) and is similar to that of the EWKM algorithm and the improved weighted fuzzy *k*-means, which changes linearly along with the number of objects, at the same time the *k*-center algorithm is used for initial *k* points, which reduces the times of iteration of *k*-means algorithm.

B. Partition of Incremental Data Points

After obtaining *k* clusters using history data points, we can classify the new coming data points into existing clusters.

In fact, we can fix the value of w_{li} by the degree of interdependencies of different attributes, which can be given through experience or according to the opinion of experts.

Given a new data point x^*, we calculate the distance between x^* and the *k* centers of existing clusters, and the l^*th cluster with smallest distance is the cluster that x^* should belong to.

$$\tau_{l^*}=\begin{cases}1, & if \sum_{i=1}^{m}\omega_{li}(c_{li}-x^*{}_i)^2 \le \sum_{i=1}^{m}\omega_{zi}(c_{zi}-x^*{}_i)^2\\ 0, & otherwise\end{cases} \tag{12}$$

Then we can adjust its new center,

$$c_{l^*i}=\frac{1}{\sum_{j=1}^{n}\tau_{l^*j}+1}\left(\sum_{j=1}^{n}\tau_{l^*j}x_{ji}+x^*{}_i\right) \tag{13}$$

Let $n=n+1$, we can classify the next new coming data points in its cluster in the same way.

The method mentioned above can be used while history data points reflect the universal set very well. But it will be not suitable for the condition that new data points with markedly dissimilar to history data points, because the existing clusters don't stand by the characteristics of new coming data points.

In many cases like web click rate, we only get data points one by one in sequence and the iteration over the whole dataset becomes impossible. Thus iteration clustering isn't suitable for partitioning incremental data points. Even if we obtain a set of data points at a time, in the case of the very large data points and the gradual process of changing, we can use one data point to stand for a set of data points, the principle of which is similar to time window.

Then, we use improved vector quantization to classify new coming data points. Extension of traditional vector quantization can be shown as (14).

$$c^{new}{}_{li}=c^{old}{}_{li}+\mu(x^*{}_i-c^{old}{}_{li}) \tag{14}$$

Here, x^* is the new coming data point, $c^{old}{}_{li}$ is the *l*th center that is closest to x^* under distance, $c^{new}{}_{li}$ is the new *l*th center after assign x^* to the *l*th cluster.

The equation (14) is similar to (13), which can not reflect distribution of the whole dataset. We use the vector quantization to partition incremental data points with existing dataset inspired by [34], which gives the extension of vector quantization for partitioning incremental data points.

When a new data point x^* is coming, we first calculate the distances between x^* and all *k* centers of clusters like (15)

$$d(x^*,c_l)=\sum_{i=1}^{m}\omega_{li}(x^*{}_i-c_{li})^2 \tag{15}$$

Here, $1\le l\le k$ and $1\le i\le m$. Given the minimum $d(x^*,c_l)$ is $d_{\min}(x^*,c_l)$ and x^* is not a faulty.

If $d_{\min}(x^*,c_l)<\rho$ then set $n=n+1$, and assign x^* to the *l*th cluster and renew its center according to (13).

Else set $k=k+1$, $n=n+1$, $c_k=x^*$

According to the method mentioned above, we know that value of ρ is critical to the clustering. We should make the parameter ρ larger to prevent the algorithm to generate too many clusters, at same time make it smaller to prevent the algorithm to generate few clusters.

The form of ρ given is given as (16) in [33].

$$\rho=\frac{0.3}{\sqrt{2}}\sqrt{m} \tag{16}$$

The 0.3 and $\sqrt{2}$ in (16) have no practical sense and they can change according to real dataset, at the same time, it is difficult to decide its valid value for different

datasets. Therefore, we give the form of ρ which is meaningful and easy to get its value in (16).

$$\rho = \frac{1}{n}\sum_{j=1}^{n}\tau_{lj}d(x_j, c_l)\times\sum_{i=1}^{m}\sqrt{\omega_i} \tag{17}$$

Here, ρ is composed of two terms, the former one is the average value of distances of n data points and centers that they belong to, and another equals to sum of square roots of weights of m dimensions.

If all dimensions have the same weights, which means that ω_i is $1/m$, (17) equals to (18)

$$\rho = \frac{\sqrt{m}}{n}\sum_{l=1}^{k}\sum_{j=1}^{n}\tau_{lj}d(x_j, c_l) \tag{18}$$

Here, the value of ρ is related to the number of attributes and the average value of inner distances.

C. Process of Fuzzy K-Means Incremental Clustering Based on K-Center and Vector Quantization

Now, we present the process of fuzzy k-means incremental clustering based on k-center and vector quantization as follows.

Step1. Eliminate the difference of units of dimensions according to (19), which will make all data points are zero dimension

$$x_{ji} = \left|\frac{x^{\text{original}}{}_{ji} - \min\limits_{t} x_{ti}}{\max\limits_{t} x_{ti} - \min\limits_{t} x_{ti}}\right| \quad 1\le i\le m \tag{19}$$

Step2. Partition initial data points into k clusters using weighted fuzzy k-means algorithm mentioned above.

Step3. Make the k means into a tree structured vector quantization using two centroids clustering, which can be shown in Figure 1.

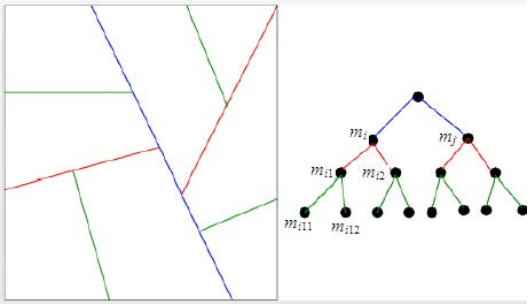

Figure1. Two centroids clustering

The act of establishing the tree can be given as follows:

1. Given the number of layers of the tree be l=1 and the number of groups g=1 for the l layer, then we calculate the mean of all k centers and make it the root of the first level of the tree and all centers are in one group.

2. Let l=l+1, we divide each of existing groups into two small groups and there may be at most 2^{l-1} groups, and let means of centers of new small groups be the roots of the l layer and each mean stands for one group for the l layer, given g be number of true groups.

3. If all groups have only one center, stop, otherwise go to 2.

In fact, we know that the all k means of the fuzzy k-means algorithm are the leaves of the tree from the process of establishing the tree.

Step4. Take the next incoming data point x^* (online case) or fetch out a data sample from a data matrix randomly or ordered (offline case), use (19) to make it dimensionless

Step5. Calculate the distance of the selected data point to root of the tree and its sub trees by using Euclidean distance as following:

Supposing the x^* is more close to m_i then to m_j, then we just need to compute two distances between the x^* and m_{i1}, and x^* and m_{i2}, supposing the distance between x^* and m_{i1} is smaller, then we need to calculate two distances between x^* and m_{i11}, and x^* and m_{i12}, and so on. If the distance between x^* and m_{i11} is smaller and m_{i11} doesn't have sub points, then x^* is closest to m_{i11}.

Step6. Elicit the mean of cluster which is closest to the data point according Step5 -> winning cluster represented by its center c_{win}.

Step7. If $d_{\min}(x^*, c_l) < \rho$ then set n=n+1

Else set k=k+1, n=n+1, c_k=x^*, go to Step9

Step8. Update the m dimensions of the winning cluster by moving it towards the selected point x^*, as in (14)

Step9. If the data matrix still contains uncovered data points (offline case) or new incoming data points are still available (online case), go to Step 4. Otherwise stop.

According to the process of fuzzy k-means incremental clustering, we know that the complexity of clustering using two centroids for the new coming data point is O (Z log$k$), here Z is the number of incremental data points, and the complexity of traditional algorithm is O (Z k). Therefore, if the number of clusters and number of incremental data points are very large, we can reduce the quantity of calculation for increment clustering in this paper.

IV. Experiment

In this section, we use Iris dataset, Indians-diabetes dataset and Zoo dataset from UCI website to validate the fuzzy k-means algorithm incremental clustering based on k-center and vector quantization. In order to see the efficiency of the algorithm, we firstly use k-means algorithm to train part of data points iteratively, then use other data points as incremental data points.

A. Iris Dataset

We start the algorithm by initialing parameters, let m =4, n=75, k=2, γ=0.5, $\omega_i = 0.25$. We choose top 75 data points as training set of k-means, which belong to two clusters, and then use fuzzy k-means algorithm to partition the rest data points. We firstly use (19) to eliminate the difference of dimensions and next use fuzzy k-mans algorithm to train these 75 data points and the value of ρ is 0.0328, then we get the results that the 94th, 110th, 118th, 119th, 123rd, 132nd data points are partitioned in to error clusters, whose error rate is 0.04 and is less than that in [34], at the same time we reduce the computation incremental mode because only half of data points were trained in the process of iteration, others are just used once.

In fact, the weights of attributes of Iris dataset, given $\omega = (0.1, 0.2, 0.3, 0.4)$, then we know that the value of ρ is 0.0345, and results show that the 110th, 118th, 119th,

132nd are in error clusters, whose error rate is 0.027, which means that we can make the degree of accuracy of the proposed algorithm higher as long as we give proper weights of attributes of datasets. As a matter of fact, the third and the forth dimensions are critical for Iris dataset. In order to see the results clearly, we give its results with the third and the forth dimensions in Figures.

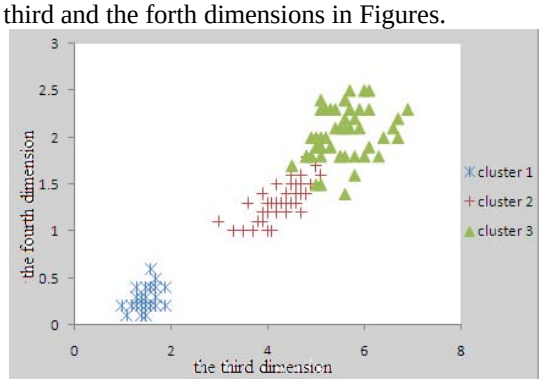

Figure2. The expecting results for Iris dataset

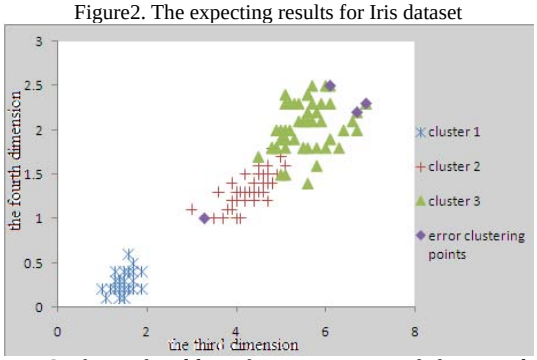

Figure3. The results of fuzzy *k*-means incremental clustering for partitioning Iris dataset

The expecting results of Iris dataset are given in Figure2 and the practical results of fuzzy k-means incremental clustering are given in Figure3. In Figure3, the error clustering points are points that partition into error clusters.

From results of classifying Iris dataset in [27], we know that accuracy of the traditional *k*-means algorithm is 88%, that of EWKM algorithm is 94.67%, that of the proposed algorithm in [27] is 95.33%, and that of fuzzy *k*-means incremental clustering is 96% and can reach to 96.55% while we give proper weights for attributes. Moreover, we reduce the quantity of data points in iteration and calculation, so it is effective and practical for classifying Iris dataset.

B. Pima-Indians-Diabetes Dataset

We begin with removing the difference of dimensions and initializing the parameters, let m =8, n=368, k=2, $\gamma=0.5$, $\omega_i = 0.125$, we use top 368 data points as training set of *k*-means, and rest of data points as incremental data points. Next, we obtain the value of ρ that is approximate 0.07. Then we get the results that the accuracy degree of the fuzzy k-means incremental clustering is 75.65% and totally 187 data points are partitioned into error clusters.

Then,

given $\omega = \{0.13,0.13,0.11,0.14,0.12,0.12,0.12,0.13\}$,

we use the fuzzy *k*-means incremental clustering, and the results are very similar to the results with the same weights of different dimensions. The accuracy degree of the fuzzy *k*-means incremental clustering is 75.91% and totally 185 data points are partitioned into error clusters.

In fact, none of dimensions of Pima-Indians-Diabetes dataset has significant influence for clustering and it is difficult to be shown in figure clearly. In Figure 4 and Figure 5, data points of both clusters mingled together and won't be divided distinctly as Iris dataset in Figure2 and Figure3.

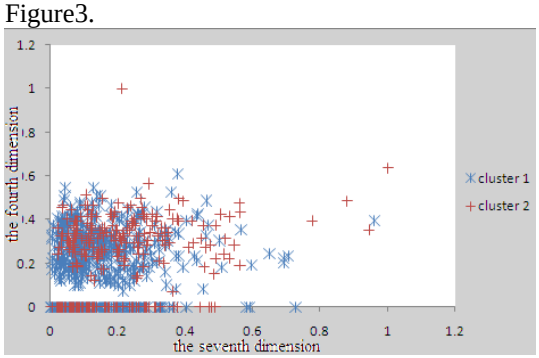

Figure4. The expecting results for Pima-Indians-Diabetes dataset

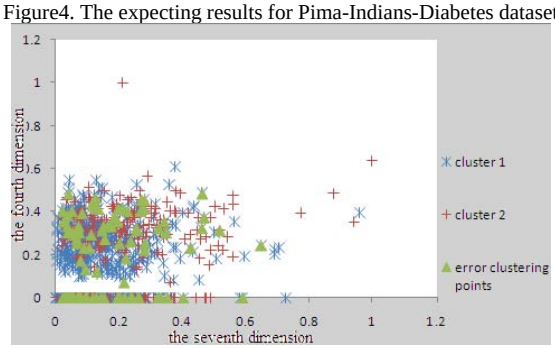

Figure5. The results of fuzzy *k*-means incremental clustering for partitioning Pima-Indians-Diabetes dataset

From the results mentioned above, we know that weights of different dimensions influence the results of clustering very little for Pima-Indians-Diabetes.

C. Segmentation Dataset

According to values of attributes of data points, we know some attributes are redundant because they won't influence the clustering like the ninth attribute, the value of which never changes or their influences for the clustering are very little, take the fourth and fifth attributes for example. In practice, we can reduce those redundant attributes through experience or special strategies or algorithms while there are too many attributes that will lead to computationally intensive.

Besides, we can remove the tenth, fourteenth, fifteenth, sixteenth, seventeenth, eighteenth and nineteenth attributes, which can be obtained from eleventh, twelve and thirteenth attributes under addition and subtraction. Then, there are only nine attributes left. We start the algorithm by removing the difference of these nine dimensions and initialing the parameters, let m =9, n=210, k=2, $\gamma=0.5$, $\omega_i = 1/9$, we choose 210 data points of all Segmentation dataset as training set of *k*-means, and next we use all of 2100 data points of Segmentation Test dataset.

Then we use all of 2100 data points as incremental data points. The accuracy degree of the fuzzy *k*-means incremental clustering is 65.86% and totally 717 data points are partitioned into error clusters for test data points.The expecting results for partitioning Segmentation Test dataset with the first and second attributes are shown in Figure6.

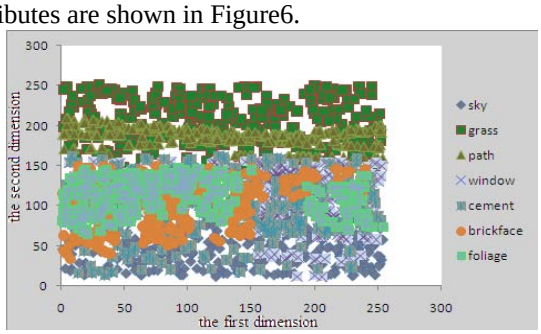

Figure6. The expecting results for all data points of Segmentation Test dataset

Now, the results of fuzzy *k*-means incremental clustering for partitioning test data points are shown in Figure9.

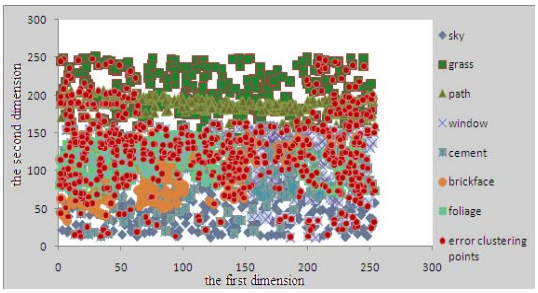

Figure7. The results of fuzzy *k*-means incremental clustering for 480 data points of Segmentation Test dataset

In Figure7, the red round points are those data points that are partitioned into wrong clusters using fuzzy *k*-means algorithm in the mode of incremental clustering.

Compared the results mentioned above with that in [34], the proposed algorithm in this paper has a higher accuracy while reduces the quantity of training data points of *k*-means and divides the rest of data points in an incremental mode, which only need calculate the distance once and remove the iteration for these data points.

V. Conclusions

Traditional fuzzy *k*-means algorithm can only be used to group data points that are given before clustering, which is ineffective while the data points are incremental or continuous. On the contrary, vector quantization is able to partition incremental data points quickly but neglects history data points.

This paper proposes an improved fuzzy *k*-means incremental clustering based on k-center and vector quantization, which connects the two methods mentioned above together, which makes use of their advantages while overcomes their drawbacks. The improved weighting entropy *k*-means clustering algorithm uses the *k*-center clustering to produce initial *k* means and makes the outside distance as large as possible while the within distance as small as possible. The improved vector quantization is used for processing incremental data points, which makes use of two centroids clustering to establish a tree for *k* means and reduce the calculation amount of incremental data points, at same time decide whether the new point should belong to an existing cluster or be a new cluster.

Finally, we use Iris dataset, Pima-Indians-Diabetes dataset and Segmentation dataset to validate the proposed algorithm. For all datasets, the proposed algorithm improves the accuracy while reduces the amount of calculation and the time of iteration, which show it is effective and feasible and will need further studies.

Acknowledgment

This work was supported by the Doctoral Fund of Ministry of Education of China (Grant No. 200801510001) and the Key Project of Chinese Ministry of Education (Grant No. 209030).

References

[1] L. Jing, M. K. Ng, and J. Z. Huang, An Entropy Weighting k-Means Algorithm for Subspace Clustering of High-Dimensional Sparse Data, *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, USA: Institute of Electrical and Electronics Engineers. 19, 8, 2007, 1026-1041.

[2] T. Zhang, R. Ramakrishnan, and M. Livny, BIRCH:An efficient data clustering method for very large databases, In *Proceedings of the Symposium on Management of Data (SIGMOD)* (Montreal, Quebec, Canada, 1996), ACM Press, New York, NY, 1996, 103-114.

[3] S. Guha, R. Rastogi, and K. Shim, CURE: An efficient clustering algorithm for clustering large databases, In *Proceedings of the Symposium on Management of Data (SIGMOD)* (Seattle, Washington, USA, 1998), ACM Press, New York, NY, 1998, 73-84.

[4] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications, In *Proceedings of the Symposium on Management of Data (SIGMOD)* (Seattle, Washington, USA, 1998), ACM Press, New York, NY, 1998, 94-105.

[5] C.H. Cheng, A.W. Fu, and Y. Zhang, Entropy-Based Subspace Clustering for Mining Numerical Data, In *Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Diego, CA, USA, August, 1999), ACM Press, 1999, 84-93.

[6] C. Aggarwal, C. Procopiuc, J. L. Wolf, P.S. Yu, and J.S. Park, Fast Algorithms for Projected Clustering, In *Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Diego, CA, USA, August, 1999), ACM Press, 1999, 61-72.

[7] C. C. Aggarwal, and P. S. Yu, Finding Generalized Projected Clusters in High Dimensional Spaces, In *Proceedings of the Symposium on Management of Data* (Dallas, Texas, USA), ACM Press, 2000, 70-81.

[8] C. C. Hsu, and Y. Huang, Incremental clustering of mixed data based on distance hierarchy, Expert Systems with Applications. 35, 3, 2008, 1177-1185

[9] J. A. Hartigan, *Clustering Algorithms*, John Wiley & Sons, Inc, New York, 1975.

[10] G. Carpenter, and S. Grossberg, Art3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures, *Neural Networks*. 3, 2, 1990, 129-152.
[11] D. Fisher, Knowledge acquisition via incremental conceptual clustering, *Machine Learning*. 2, 1987, 139-172.
[12] F. Can, Incremental clustering for dynamic information processing. *ACM Transaction for Information Systems*. 11 1993, 143-164.
[13] F. Can, E. A. Fox, C. D. Snavely, and R. K. France, Incremental clustering for very large document databases: Initial MARIAN experience. *Inf. Sci*. 84, 1995, 101-114.
[14] T. Langford, C. G. Giraud-Carrier, and J. Magee, Detection of infectious outbreaks in hospitals through incremental clustering, In *Proceedings of the 8th Conference on AI in Medicine* (Cascais, Portugal, July 1-4, 2001), Springer, Berlin, 2001, 30-39.
[15] J. Lin, M. Vlachos, E. J. Keogh, and D. Gunopulos, Iterative Incremental clustering of time series, *Lecture notes in computer science*, 2004, 106-122.
[16] M. Charikar, C. Chekuri, T. Feder, and R. Motwani, Incremental clustering and dynamic information retrieval, In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, 1997, 626-635.
[17] M. Ester, H. P. Kriegel, J. Sander, M. Wimmer, and X. Xu, Incremental clustering for mining in a data warehousing environment, In *Proceedings of the 24rd International Conference on Very Large Data Bases* (New York City, New York, USA, August 24-27, 1998), Morgan Kaufmann, 1998, 323-333.
[18] D. Simovici, N. Singla, M. Kuperberg, and M. Kuperberg, Metric Incremental Clustering of Nominal Data, In *Proceedings of the 4th IEEE International Conference on Data Mining* (Brighton, UK, Nov 2004.), IEEE Computer Society, 2004, 523-527.
[19] A. Jain, and R. Dubes, *Algorithms for clustering Data*, Englewood Cliffs, NJ: Prentice Hall College Div, 1988.
[20] A. Ahmad, and L. Dey, A k-mean clustering algorithm for mixed numeric and categorical data, *Data & Knowledge Engineering*, 63, 2007, 503-527.
[21] J.Z. Huang, M.K. Ng, H. Rong, and Z. Li, Automated Variable Weighting in k-Means Type Clustering, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27, 5, 2005, 1-12.
[22] H. Friguiand, and O. Nasraoui, Unsupervised Learning of Prototypes and Attribute Weights, *Pattern Recognition*, 37, 3, 2004, 567-581.
[23] Y. Chan, W. Ching, M. K. Ng, and J.Z. Huang, An Optimization Algorithm for Clustering Using Weighted Dissimilarity Measures, *Pattern Recognition*, 37, 5, 2004, 943-952.
[24] C. Domeniconi, *Locally Adaptive Techniques for Pattern Classification*, PhD dissertation, University of California, 2002.
[25] C. Domeniconi, D. Papadopoulos, D. Gunopulos, and S. Ma, Subspace Clustering of High Dimensional Data, In *Proceedings of SIAM International Conference on Data Mining* (Florida, USA, April 22-24, 2004), 2004, 517-520.
[26] S. Yang, Y. Li, X. Hu, and R. Pan, Optimization Study on k Value of K-means Algorithm, *Systems Engineering-theory & Practice*. Beijing, China, 27, 2, 2006, 97-101.
[27] T. Li, and Y. Chen, An Improved k-means Algorithm for Clustering Using Entropy Weighting Measures, In *Proceedings of the 7th World Congress on Intelligent Control and Automation (WCICA 2008)* (Chongqing, China, Jun 25-27, 2008), IEEE, 2008, 149-153.
[28] G. L. Somlo, and A. E. Howe, Incremental Clustering for Profile Maintenance in Information Gathering Web Agents, In *Proceedings of the fifth international conference on Autonomous agents* (Montreal, Canada, May 28-June 1, 2001), ACM Press, 2001, 262-269.
[29] N. Chen, A. Chen, and L. Zhou, An incremental grid density-based clustering algorithm, *Journal of Software*. Beijing, China, 13, 1, 2002, 1-7.
[30] Y. Huang, and L. Zou An incremental density-based clustering algorithm in a batch mode used in a data warehouse, *Computer Engineering and Applications*. Beijing, China, 29, 2004, 206-208.
[31] X. Xu, and Y. Xie, Summarization on incremental clustering and research of incremental DBSCAN algorithm, *Journal of North China Institute of Astronautic Engineering*. Langfang, Hebei, China, 16, 2, 2006, 15-17.
[32] J. Liu, and F. Li, An Efficient Incremental Algorithm for Clustering Based on Density, *Computer Engineering*, Shanghai, China, 32, 21, 2006, 76-78.
[33] E. Lughofer, Extensions of vector quantization for incremental clustering, *Pattern Recognition*. 41, 2008, 995-1011.
[34] X. Qian, X. Huang, and L. Wu, A Spectral Method of K-means Initialization. *ACTA AUTOMATICA SINICA*, Beijing, China, 33, 4, 2007, 342-346.

Taoying Li was born in Anhui Province, China on September 1983. Taoying Li received the BE degree in information management and information system at Dalian Maritime University, Dalian, China, in 2005 She is currently working toward the Successive Postgraduate and Doctoral Progr- am in the Transportation Management College, Dalian Maritime University.

She had carried out some projects and published several papers. She majors in Management science and Engineering and her research interests include data mining, system engineering, and artificial intelligence.

Yan Chen was born in Liaoning Province, China on December 1952. Yan Chen received the BE degree in computer software at Dalian Maritime College, Dalian, China, in 1978, the MS degree in computer application at Dalian Maritime College in 1989 and the PhD degree in Management Science and En-gineering at Dalian University of Technology in 2000.

She is a professor in Transportation Management College, Dalian Maritime University, and the dean of the Key Laboratory in Liaoning Province on Logistics Shipping Management System Engineering. She has published three books and published more than 100 papers. Her main research interests include data mining, system engineering, special data mining, decision-making support system and data warehouse.