An Improved K-means Algorithm Based on Structure Features

Qiang Zhan*
School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China.
College of Engineering, Forestry, and Natural Sciences, Northern Arizona University, Arizona, America.

* Corresponding author. Tel.: 86-15035054991; email: ZQ156259@126.com
Manuscript submitted September 12, 2016; accepted December 20, 2016.
doi: 10.17706/jsw.12.1.62-81

Abstract: In *K*-means clustering, we are given a set of *n* data points in multidimensional space, and the problem is to determine the number *k* of clusters. In this paper, we present three methods which are used to determine the true number of spherical Gaussian clusters with additional noise features. Our algorithms take into account the structure of Gaussian data sets and the initial centroids. These three algorithms have their own emphases and characteristics. The first method uses Minkowski distance as a measure of similarity, which is suitable for the discovery of non-convex spherical shape or the clusters with a large difference in size. The second method uses feature weighted Minkowski distance, which emphasizes the different importance of different features for the clustering results. The third method combines Minkowski distance with the best feature factors. We experiment with a variety of general evaluation indexes on Gaussian data sets with and without noise features. The results showed that the algorithms have higher precision than traditional *K*-means algorithm.

Key words: *K*-means, feature weighting, clustering, cluster validity index.

1. Introduction

Clustering is an effective tool for data mining, and it has been widely concerned by people. The previous classification algorithm is to divide the data into a pre-labeled class. But in some cases, we need to analyze a data set without knowing the structure and distribution of the data in advance, so that the classification algorithms have no ability to handle the data in this way, because the classification algorithms are supervised [1]. Then we can use the clustering algorithms to analyze the data.

Clustering can help people analyze data and solve practical problems. In the fields of biological information processing, psychological research, business data analysis, network data analysis, geography and information retrieval, clustering has a wide range of applications [2]–[4].

Clustering is a classical method of unsupervised data analysis. Up to now, there is no universally accepted clustering definition of the academic circle. Here, we quote the definition of the clustering which was given by the Everitt [5] in 1974: "It makes the data objects in the same cluster have high similarity, and the data objects in different clusters are not similar". Cluster analysis is a technique that does not require prior knowledge of the class label.

The existing clustering methods are divided into hierarchical clustering [6]–[10] , partition clustering [11] –[29], and Clustering algorithm based on grid and density [30]–[34].

In the original *K*-Means, this similarity measure is the squared Euclidean distance. There are other

distance similarities, such as the Jaccard [35], Cosine [35], and Edit [36] similarity. Cosine distance, also known as cosine similarity, is used to measure the angle between two vectors in the vector space. It serves as a measure of the size of the difference between the two individual. Euclidean distance and cosine distance have different computing methods and characteristics, so they can be applied to different data analysis models: Euclidean distance can reflect the absolute difference among the individual numerical characteristics. Euclidean distance is used more to reflect the differences from the dimension sizes. Cosine distance is used more to reflect the differences from the direction, and is not sensitive to the absolute value, therefore Cosine distance is not appropriate for data sets with Gaussian distribution.

Among all kinds of clustering algorithms, the *K*-means algorithm based on partition is widely used because of its simplicity and its ability to effectively cluster large data sets. The clustering results of the *K*-means algorithm are very sensitive to the initial center point, and the improper initial cluster centers will lead to the instability of the cluster structure. At the same time, the algorithm is sensitive to the data dimension, because the similarity measure of the algorithm is the Euclidean distance, and the importance of all features in the calculation of the distance is the same. This processing method, which is not distinguished from the importance of attributes, is likely to result in the distance distortion of the data points in the space. If the two points in the space are very close on the important features, but due to the amplification of the distance by other irrelevant features, these two points in the Euclidean space are likely to be the most measured. Because the Euclidean distance is relatively simple, and can basically reflect the performance of the algorithm, therefore, it was more commonly used. In the cluster analysis, the distance is not fixed. Other distances are also useful, and clustering algorithms can use different distances according to the specific problems. For example, we can use Mahalanobis distance to increase the recognition ability of the cluster structure of ellipsoidal shape. Mahalanobis distance is an extension of Euclidean distance and its equidistant points are composed of a hyper ellipsoid, while the equidistant points of Euclidean distance form a sphere. Another example, the distance function using exponential form can suppress noise.

Clustering evaluation index is an essential part in the process of clustering, we can evaluate the advantages and disadvantages of the clustering algorithm through the cluster indicators. For example, the NMI index is calculated by comparing all the data mutual information of the two clusters to compare the similarity of the two clusters. But so far, there is not a common index can be applied to all the data sets and all the algorithms. Different clustering validation indexes can be applied to different clustering tasks. When performing a cluster analysis, we first have to determine what kind of clustering task we are going to perform. In this paper, we use a variety of general clustering validity indexes to test our algorithms.

For the above defects of the *K*-means algorithm, the main contributions of this paper lie in the following three aspects:

1) For concern that the K-means algorithm is suitable for the initial center points, a method choosing the initial center points is proposed.
2) In view of the low noise suppression ability of K-means algorithm, a new algorithm based on feature importance is proposed.
3) An algorithm combining the feature importance and the optimal feature is proposed.

K-means and several validation indexes are reviewed in Section 2. Section 3 describes our algorithms. In Section 4 we present our simulation and analysis of the results. The conclusion is given in Section 5.

2. Background

2.1. *K*-means

The procedure of the *K*-means algorithm is as follows.

Distribute all objects to *k* number of different cluster at random; calculate the mean value of each cluster,

and use this mean value to represent the cluster; re-distribute the objects to the closest cluster according to its distance to the cluster center; update the mean value of the cluster, then calculate the mean value of the objects in each cluster; and calculate the criterion function until the criterion function converges. Usually, the *K*-mean algorithm criterion function adopts square error criterion, and is defined as

$$\mathrm{E} = \sum_{i=0}^{k} \sum_{P \in C_i} |p - m_i|^2 \tag{1}$$

In which E is total square error of all the objects in the data cluster, p is the given data object, m_i is mean value of cluster C_i (p and m are both multi-dimensional). The function of this criterion is to make the generated cluster as compacted and independent as possible.

2.2. Silhouette Index

Silhouette index [37] was first described by Peter J. Rousseeuw in 1986.

Assuming $a(t)$ is the average similarity or distance between the sample t in the clustering C_j and all other samples in the same clustering, $d(t, C_i)$ is the average similarity or distance between the sample t and all the samples in another clustering C_i, then $b(t) = min\{d(t, C_i)\}, i = 1,2, \dots k, i \neq j$. *Sil* indicators calculate the distance between each sample and other samples in the same cluster, and the distance between the sample in other clusters, and the calculation formula of each sample t is as follows:

$$Sil(t) = \frac{b(t) - a(t)}{max\{a(t), b(t)\}} \tag{2}$$

The formula can be also written as :

$$Sil(t) = \begin{cases} 1 - \dfrac{a(t)}{b(t)}, \text{if } a(t) < b(t) \\ 0, \qquad \text{if } a(t) = b(t) \\ \dfrac{b(t)}{a(t)} - 1, \text{if } a(t) > b(t) \end{cases} \tag{3}$$

From the above definition, it is clear that:

$$-1 \leq Sil(t) \leq 1 \tag{4}$$

In general, the average Sil value of all samples in a data set is used to evaluate the quality of the clustering results. The greater the Sil index, the better the quality of the clustering. The maximum value of the Sil index corresponding to the number of clusters is regarded as the optimal number of clusters.

2.3. Davies-Bouldin(DB) Index

DB index [38] is based on the class scatter of samples and the measure of the distance between the cluster centers. The number of the class corresponding to the minimum is the optimal number of clusters.

Set DW_i to represent the average distance between all the samples in the cluster C_i and their cluster centers, DC_{ij} indicates the distance between the cluster C_i center and the C_j center, then the DB index is as follows :

$$DB(k) = \frac{1}{k} \sum_{i=1}^{k} \max_{j=1 \sim k, j \neq i} \left(\frac{DW_i + DW_j}{DC_{ij}}\right) \tag{5}$$

2.4. Weighted Inter Intra Similarity Ratio (Wint)

The goal of Wint is to maximize the similarity within the class and to minimize the similarity between classes.

Usually, the Wint index with penalty term is used to estimate the class number, and its maximum value is regarded as the optimal number of clusters.

Wint index is defined as:

$$Wint(k) = 1 - \frac{1}{\sum_{i=1}^{k} C_i \times intra(i)} \sum_{i=1}^{k} \frac{n_i}{n - n_i} \sum_{j=1, j \neq i}^{k} n_j \times intra(i,j) \tag{6}$$

$$intra(i) = \frac{2}{n_i(n_i - 1)} \sum_{s,t \in C_i, s<t} R(s,t) \tag{7}$$

$$inter(i,j) = \frac{1}{n_i n_j} \sum_{s \in C_i, t \in C_j} R(s,t) \tag{8}$$

2.5. Homogeneity-Separation (HS) Index

HS index uses homogeneity to represent the cohesive structure of samples in a cluster, and uses separation to represent classes that are well separated from each other.

Homogeneity is defined as the average within class samples between similarity and separability is defined as the average similarity between samples of different classes.

The class number corresponding to the maximum value of the *HS* index is the optimal number of clusters. Its definition is:

$$HS(k) = |Hom(k) - Sep(k)| \tag{9}$$

In this equation,

$$Hom(k) = \frac{2}{\sum_{i=1}^{k} n_i(n_i - 1)} \sum_{i=1}^{k} \sum_{s,t \in C_i, s<t} R(s,t) \tag{10}$$

$$Sep(k) = \frac{2}{\sum_{i,j=1; i<j}^{k} n_i n_j} \sum_{i,j=1; i<j}^{k} \sum_{s \in C_i, t \in C_j} R(s,t) \tag{11}$$

In this equation, $R(s,t)$ represents the similarity between the sample s and t.

2.6. Dunn's Index

Dunn's index [39] is defined as the ratio of the smallest distance between clusters, which estimates the separation of clusters, and the maximum cluster diameter, which estimates its cohesion. This index allows for general distance measures.

The definition of *Dunn* is as follows:

$$D_{n_c} = \min_{i=1,\ldots,n_c} \left\{ \min_{j=i+1 \ldots n_c} \left(\frac{d(c_i, c_j)}{\max_{k=1 \ldots n_c} diam(c_k)} \right) \right\} \tag{12}$$

In which, $d(c_i, c_j)$ is the inconsistency measure between cluster c_i and cluster c_j, which is the minimum distance between the two clusters. $Diam(c)$ is the diameter of the cluster c, which can be used to measure the degree of dispersion of the data in the cluster. The definition of Diam is as follows:

$$Diam(c) = \max_{x,y \in c} d(x, y) \tag{13}$$

The greater the Dunn value, the better the effect of clustering.

3. Proposed Method

Before KMENAS clustering, the initial center is selected firstly, and then the initial partition is carried on, however, the initial value of the selection and clustering results have a great association. If the choice of the initial center is poor, it may produce invalid clustering results. At the same time, the result of the clustering is different, which leads to the poor stability of the clustering results. Secondly, each cluster center is the average of all samples in each cluster, using the Euclidean clustering as the similarity metric, and the use of square error function as the criterion function clustering. This will cause that the globular clusters can be easily found, but when the difference of the cluster size and shape is relatively large, it is not easy to be found. There may be large clusters are segmentation, because the criterion function to achieve optimal results. Finally, because the algorithm uses the mean of all objects of each cluster as the cluster center, however, outlier data generally deviate from the cluster center, so it will bring greater effect on the calculation of mean. This will cause the deviation of the clustering center and inaccuracy of the clustering results.

3.1. Relevant Definition

3.1.1. Preliminary

1) The $n \times m$ dimension objects are represented as following matrix form:

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix} \tag{14}$$

In order to make the data of different features comparable, and in order to calculate the contribution degree of features, the matrix is normalized according to the dimension [0.01, 1].

2) After the current iteration, the n objects is divided into K clustering. The number of objects in each cluster is $\{n_1, n_2 \ldots n_k\}$. Then the sum of the distance within a class of all K cluster on the feature j is:

$$\mathrm{d_n} = \sum_{\mathrm{k=1}}^{\mathrm{K}} \sum_{\mathrm{i=1}}^{\mathrm{n_k}} (\mathrm{x_{ij}} - \mathrm{m_{kj}})^{\mathrm{p}} \tag{15}$$

m_{kj} is the mean value of clustering k on the feature j.

3) The sum of the distance within the class of all K clustering on the feature j is:

$$\mathrm{d_w} = \sum_{\mathrm{k=1}}^{\mathrm{K}} (\mathrm{m_{kj}} - \mathrm{m_j})^{\mathrm{p}} \tag{16}$$

m_j is the mean value of data set on the feature j.

4) According to the current iteration results, the contribution of feature j to the cluster is calculated:

$$c_j = \frac{d_w}{d_n} \tag{17}$$

Compactness and separation are usually used to measure the overall performance of clustering.

For a single attribute, if the result of the clustering is that data objects within a class are compact and the data objects between classes are separative, then the ability of the feature to distinguish objects is strong, and the contribution to clustering is large. On the other hand, the contribution of the feature to clustering is small.

3.1.2. Feature adjustment

The feature weights of the feature j is:

$$w_j = \frac{c_j}{\sum_{j=1}^{m} c_j}, w_j \in [0,1], \sum_{j=1}^{m} c_j = 1 \tag{18}$$

Using the modified Minkowski distance formula, the weighted Minkowski distance formula is obtained:

$$d(m,n) = \sqrt[p]{\sum_{j=1}^{m} w_j (x_{mj} - x_{nj})^p} \tag{19}$$

Feature weights are calculated according to the contribution of each attribute.

The larger the feature weight, the more important the feature is, and the coordinate axis of the feature in the multidimensional space should be extended; The smaller the feature weight, the smaller effect the feature is, and the coordinate axis of the attribute in the Minkowski space should be reduced.

Compared with the Minkowski space, the regulation of w can be used to cluster in the subspace, which can reflect the distribution of data set in Minkowski space, and improve the performance of clustering.

3.1.3. Selection of initial cluster centers

The *K*-means algorithm is sensitive to the initial clustering center and the different clustering centers lead to different clustering results. The traditional *K*-means algorithm does not consider the importance of each feature; there will be the distance distortion phenomenon. The selection of the initial cluster center does not completely describe the class structure of the data set. In order to improve the performance of the traditional *K*-means clustering algorithm, this paper put forward the following methods to determine the initial center point and determine in the center points of each features of the objects respectively. The center points of the entire data set are obtained through the center point of each features. The procedure is as follows: first, calculate the mean value in each features of the objects and all the variance, variance reflects the feature values with respect to discrete degree of mean value, then construct offset factor through the mean square deviation and the cluster number K. Finally, according to the mean and deviation factor, obtain initial cluster center. The algorithm is as follows:

$$\begin{cases} \left\{mean \pm \frac{2\sigma}{k-1} \times j, j = 1, \dots, \frac{k}{2}\right\} \cup \{mean\}, & k \in Odd\ numbers \\ \left\{mean \pm \frac{2\sigma}{k} \times j, j = 1, \dots, \frac{k}{2}\right\} & k \in Even\ numbers \end{cases} \tag{20}$$

$\frac{2\sigma}{k}$ and $\frac{2\sigma}{k-1}$ in the above formula are called offset factors.

3.2. Minkowski Weighted *K*-Means (MWK-Means)

The MWK-Means [40,41] algorithm is an extension of the *K*-means algorithm to solve the main shortcomings of the *K*-means algorithm, meaning that the algorithm can not effectively deal with the features of the data set. In the original *K*-means algorithm, the *V* dimensional feature is considered to be important for clustering, which means that they have the same effect on clustering. In fact, even if there are two related features, they also have a different degree of correlation. These should be taken into account by the algorithm.

The Minkowski metric, defined as $d(y_i, c_i) = \sqrt[p]{\sum_{v=1}^{m}(y_{iv} - c_{kv})^p}$ for the m-dimensional y_i and c_k, is a generalization of the Manhattan ($p = 1$) [42], Euclidean ($p = 2$) [43] and Chebyshev ($p \to \infty$) [44] metrics. The MWK-Means actually uses the pth power of the Minkowski metric.

3.3. Feature Adjusted Minkowski Weighted *K*-Means (FAMWK-Means)

We combine *K*-means with the Feature Adjustment Minkowski distance and initial cluster centers for providing good initial centroids.

The procedure of FAMWK-Means is as follows:

1) Calculate mean and mean square deviation of each attribute of the object.
2) Construct initial cluster center C according to formula (20).
3) Initialize feature weight according to formula (17).
4) Run K-means algorithm based on weighted Minkowski distance is run, and assign the cluster number to each object. Then the clustering center is re-calculated.
5) To determine whether there is a number of 0 of the cluster; if there is, the data object distribution is very dense, and the migration factor is too large. Will be offset by half, to re select the cluster center.
6) According to the result of iteration, the characteristic weight of each attribute is adjusted according to the formula.
7) Repeated execution of 4) and 6) until a predetermined number of iterations is reached or each cluster is not changed.

3.4. FAMWK-Means with Best Feature Adjustment

Our third and last method takes the previous ideas even further. We now take into account that although *K*-means applies feature weights, these are not optimized in the beginning of the clustering process. This means that in most cases the optimal weights found by *K*-means are only used in the clustering in its very last iteration, while suboptimal weights are used in all previous iterations. For this reason, we proposed to re-cluster Y with the final weights from *K*-means using a fully iterated *K*-means.

The procedure of this method is as follows:

1) Calculate mean and mean square deviation of each attribute of the object.
2) Construct initial cluster center C according to formula (20).
3) Set feature weight according as the optimal features.
4) Run K-means algorithm based on weighted Minkowski distance is run, and assign the cluster number to each object. Then the clustering center is re-calculated.
5) Determine whether there is a number 0 of the cluster. If there is, the data object distribution is very dense, the migration factor is too large. Will be offset by half, to re select the cluster center.
6) According to the result of iteration, the characteristic weight of each attribute is adjusted according to the formula.
7) Repeated execution (4) and (6) until a predetermined number of iterations is reached or each cluster is not changed.

4. Implementation

4.1. Data Set Preparation and Evaluation Standard

The implementation is done by Matlab. The code is run on a Intel(R) Core(TM) I3 CPU M 380 @ 2.53, 3.00GB Installed memory, 64 bit OS system.

In this paper, we generated 600 data sets that obey the Gauss distribution. Each Gauss data set is subject to a distribution of variance of 0.5. We use 9 different configurations. Noise features are selected to obey the uniform distribution of white noise, the mean is 0.5, and the variance is 0.5. Specific configuration is shown as Table 1:

Table 1. Different Configurations of Experimental Data Sets

	Entities	features	clusters	noise features	sum of total features
600×6-2	600	6	2	0	6
600×6-2+3NF	600	6	2	3	9
600×6-2+6NF	600	6	2	6	12
600×6-3	600	6	3	0	6
600×6-3+3NF	600	6	3	3	9
600×6-3+6NF	600	6	3	6	12
600×12-2	600	12	2	0	12
600×12-2+6NF	600	12	2	6	18
600×12-2+12NF	600	12	2	12	24
600×12-3	600	12	3	0	12
600×12-3+6NF	600	12	3	6	18
600×12-3+12NF	600	12	3	12	24

In Table 1, 600×6-2 represents that this data set contains 1000 entities and 6 features, and these entities are divided into 2 clusters. 600×6-2+3NF means that this data set contains 1000 entities, 6 features and 3 noise features, and these entities are divided into 2 clusters. The following configurations are also similar. Each configuration contains 50 randomly generated data sets.

1) F-measure index.

It is the combination of precision and recall. They are defined as follows:

$$P = precision(i,j) = \frac{N_{ij}}{N_i} \tag{21}$$

$$R = recall(i,j) = \frac{N_{ij}}{N_j} \tag{22}$$

Among them, N_{ij} is the number of j in cluster i, N_j is the number of all the objects in the cluster N_i; j is the number of all the objects in i. The F-measure of the clustering is defined as follows:

$$F(i) = \frac{2PR}{P+R} \tag{23}$$

2) Rand index

Assuming a clustering result of data set X is $C = \{C_1, C_2, \dots, C_m\}$, and the data set is known to be divided into $P = \{P_1, P_2, \dots, P_s\}$. we can evaluate he quality of clustering by comparing the C and P as well as the adjacent matrix. For each point (X_v, X_u) in the data set, calculate the following items:

SS: If two points belong to the same cluster in P and C;

SD: If two points belong to the same cluster in C, and belong to different clusters in P;

DS: If two points belong to the different cluster in C, and belong to same clusters in P;

DD: If two points belong to the different cluster in C and P;

Let a, b, c and d represent the number of SS, SD, DS and DD respectively, then $M = a + b + c + d$ represents the maximum number of all pairs in the data set, that is, $M = \frac{N(N-1)}{2}$.

In this equation, N is the total number of data points. The similarity between C and P can be defined by the following validity index:

$$R = \frac{(a+d)}{M} \tag{24}$$

4.2. Results and Analysis

Our first group of experiments run on the data sets with no noise features. We used *K*-means (baseline), MWK-MEANS, FAMWK-Means and FAMWK-Means with Best Feature Adjustment to carry out the experiments. The results are shown in the Table 2 and Table 3.

The values of *p* in table 2 have two meanings in our experiments. Firstly, they mean that our clustering algorithms, as we describe above, use the *p* values to carry out clustering. Secondly, in cluster validity indexes of Silhouette, they are also used to evaluate the quality of the clustering results. For example, in Table 2, the entries under *p*=1.6 means that clustering algorithm was run with this particular p. The results presented for the Silhouette (Mink) in row *p*=1.6 also use the same *p*=1.6 for the distance used within the index calculation.

In each table the row labeled $p \to 1$ and $p \to 3$ presents the results for a p very close to 1 (1.0001) and 3 (2.9999). We have not experimented with $p \to 1$ and $p \to 3$ because they may cause empty cluster.

Table 2 shows *K*-means and MWK-MEANS methods produce better results. *K*-means has achieved the maximum value of 0.968; MWK-MEANS obtains the best value of this experiment at *p*=2 by Silhouette using the Euclidean distance. But we noticed that compared with the *K*-means, MWK-MEANS cannot improve the F-measure with most of pth values because these data sets have been generated with the same Gaussian model, which means that each features has the same degree of relevance. It can be inferred that *K*-means and MWK-MEANS methods can be applied to noise-free data sets with similar processing capability.

Table 3 gives the Rand index. Table 3 shows that when p=2, the RAND index has achieved the maximum value. At the same time, most of the P values did not achieve better results than the *K*-means method. This shows that the data set has the same relevance with the data set of Gauss distribution. And for Gauss distribution Euclidean distance has a good clustering effect, so *K*-means and MWK-MEANS with p=2 can achieve good results. MWK-MEANS has achieved the maximum value of 0. 961 by Silhouette using the Euclidean distance.

Table 2. F-measure of Different Methods Applied in Data Sets with No Noise

	Silhouette						
	Eucl	Manh	Mink	DB	Wint	HS	Dunn
K-Means	0.968	0.959	0.968	0.927	0.962	0.947	0.951
MWK-MEANS							

p→1	0.951	0.948	0.951	0.849	0.935	0.937	0.939
p=1.1	0.955	0.949	0.955	0.853	0.938	0.939	0.940
p=1.2	0.959	0.951	0.959	0.858	0.938	0.941	0.943
p=1.3	0.959	0.951	0.959	0.858	0.943	0.940	0.942
p=1.4	0.960	0.952	0.960	0.891	0.946	0.941	0.943
p=1.5	0.962	0.952	0.962	0.871	0.946	0.941	0.942
p=1.6	0.962	0.956	0.962	0.866	0.949	0.945	0.946
p=1.7	0.965	0.955	0.965	0.810	0.951	0.944	0.946
p=1.8	0.966	0.956	0.966	0.812	0.954	0.945	0.946
p=1.9	0.968	0.959	0.968	0.819	0.956	0.948	0.950
p=2	0.969	0.962	0.969	0.824	0.956	0.950	0.953
p=2.5	0.833	0.829	0.833	0.939	0.959	0.818	0.819
p→3	0.736	0.732	0.736	0.943	0.959	0.721	0.723
			FAMWK-Means				
p→1	0.940	0.936	0.939	0.837	0.924	0.924	0.928
p=1.1	0.943	0.938	0.943	0.842	0.926	0.926	0.928
p=1.2	0.948	0.940	0.948	0.847	0.925	0.931	0.931
p=1.3	0.948	0.939	0.946	0.846	0.931	0.929	0.932
p=1.4	0.949	0.940	0.948	0.878	0.934	0.929	0.930
p=1.5	0.951	0.941	0.950	0.860	0.935	0.930	0.929
p=1.6	0.949	0.946	0.951	0.855	0.939	0.933	0.934
p=1.7	0.955	0.944	0.953	0.798	0.939	0.934	0.935
p=1.8	0.955	0.944	0.953	0.800	0.943	0.933	0.933
p=1.9	0.958	0.947	0.957	0.807	0.946	0.936	0.938
p=2	0.957	0.949	0.957	0.812	0.944	0.937	0.941
p=2.5	0.821	0.816	0.822	0.927	0.947	0.806	0.808
p→3	0.723	0.721	0.724	0.932	0.947	0.709	0.710
			FAMWK-Means with Best Feature Adjustment				
p→1	0.945	0.939	0.944	0.841	0.927	0.930	0.932
p=1.1	0.947	0.941	0.946	0.847	0.929	0.931	0.932
p=1.2	0.952	0.943	0.951	0.851	0.930	0.934	0.935
p=1.3	0.950	0.943	0.952	0.851	0.936	0.932	0.935
p=1.4	0.953	0.944	0.953	0.885	0.939	0.933	0.937
p=1.5	0.954	0.945	0.954	0.863	0.939	0.934	0.935
p=1.6	0.956	0.949	0.956	0.857	0.942	0.937	0.939
p=1.7	0.958	0.947	0.957	0.803	0.944	0.935	0.940
p=1.8	0.959	0.948	0.957	0.803	0.947	0.938	0.938
p=1.9	0.961	0.953	0.960	0.813	0.948	0.941	0.942
p=2	0.961	0.956	0.963	0.815	0.950	0.943	0.946
p=2.5	0.825	0.822	0.825	0.931	0.952	0.811	0.812
p→3	0.730	0.725	0.728	0.936	0.951	0.714	0.715

Table 3. Rand Index of Different Methods Applied in Data Sets with No Noise

	Silhouette						
	Eucl	Manh	Mink	DB	Wint	HS	Dunn
K-Means	0.957	0.943	0.957	0.927	0.941	0.938	0.939
			MWK- Means				
p→1	0.939	0.938	0.942	0.840	0.925	0.929	0.929
p=1.1	0.942	0.941	0.943	0.844	0.926	0.928	0.931
p=1.2	0.949	0.941	0.948	0.845	0.928	0.928	0.932
p=1.3	0.946	0.941	0.946	0.849	0.935	0.931	0.929
p=1.4	0.950	0.942	0.948	0.878	0.937	0.929	0.931
p=1.5	0.950	0.940	0.953	0.859	0.938	0.932	0.931
p=1.6	0.951	0.944	0.952	0.854	0.938	0.937	0.937
p=1.7	0.956	0.942	0.956	0.801	0.942	0.935	0.938
p=1.8	0.956	0.945	0.953	0.802	0.943	0.932	0.938
p=1.9	0.957	0.951	0.958	0.811	0.947	0.940	0.941
p=2	0.961	0.951	0.961	0.812	0.944	0.941	0.941
p=2.5	0.822	0.817	0.821	0.928	0.949	0.805	0.806
p→3	0.726	0.719	0.724	0.933	0.947	0.712	0.715
			FAMWK- Means				
p→1	0.927	0.925	0.931	0.825	0.911	0.915	0.916
p=1.1	0.932	0.928	0.935	0.831	0.914	0.914	0.917
p=1.2	0.936	0.928	0.939	0.835	0.915	0.920	0.918

p=1.3	0.938	0.928	0.938	0.833	0.919	0.916	0.921
p=1.4	0.936	0.929	0.937	0.868	0.923	0.918	0.922
p=1.5	0.941	0.929	0.940	0.846	0.922	0.917	0.919
p=1.6	0.938	0.932	0.938	0.842	0.928	0.923	0.921
p=1.7	0.942	0.935	0.942	0.785	0.929	0.919	0.926
p=1.8	0.945	0.934	0.943	0.791	0.933	0.923	0.922
p=1.9	0.945	0.935	0.948	0.795	0.936	0.925	0.930
p=2	0.946	0.940	0.948	0.799	0.936	0.929	0.930
p=2.5	0.810	0.804	0.811	0.918	0.935	0.795	0.798
p→3	0.715	0.709	0.715	0.922	0.937	0.699	0.699
FAMWK- Means with Best Feature Adjustment							
p→1	0.930	0.926	0.930	0.828	0.912	0.916	0.917
p=1.1	0.930	0.927	0.934	0.830	0.915	0.917	0.917
p=1.2	0.938	0.930	0.939	0.837	0.914	0.919	0.922
p=1.3	0.936	0.927	0.935	0.835	0.921	0.920	0.919
p=1.4	0.936	0.929	0.939	0.869	0.924	0.916	0.920
p=1.5	0.937	0.931	0.942	0.848	0.924	0.920	0.922
p=1.6	0.940	0.931	0.938	0.842	0.926	0.923	0.922
p=1.7	0.943	0.932	0.942	0.787	0.930	0.922	0.922
p=1.8	0.944	0.935	0.945	0.787	0.934	0.924	0.923
p=1.9	0.947	0.936	0.946	0.795	0.932	0.924	0.928
p=2	0.948	0.939	0.949	0.803	0.933	0.928	0.933
p=2.5	0.810	0.808	0.809	0.915	0.938	0.794	0.795
p→3	0.712	0.712	0.712	0.922	0.938	0.699	0.702

Table 4. F-measure of Different Methods Applied in Data Sets with 50% Noise

	Silhouette						
	Eucl	Manh	Mink	DB	Wint	HS	Dunn
K-Means	0.823	0.767	0.821	0.721	0.829	0.793	0.866
MWK- Means							
p→1	0.890	0.888	0.891	0.784	0.872	0.872	0.876
p=1.1	0.894	0.885	0.894	0.788	0.877	0.874	0.876
p=1.2	0.895	0.887	0.896	0.796	0.875	0.878	0.882
p=1.3	0.895	0.891	0.898	0.798	0.880	0.875	0.879
p=1.4	0.900	0.892	0.896	0.827	0.883	0.879	0.880
p=1.5	0.898	0.890	0.899	0.811	0.885	0.880	0.881
p=1.6	0.900	0.891	0.901	0.806	0.885	0.885	0.886
p=1.7	0.904	0.892	0.900	0.747	0.891	0.882	0.884
p=1.8	0.903	0.893	0.905	0.751	0.893	0.884	0.884
p=1.9	0.903	0.897	0.908	0.759	0.892	0.886	0.885
p=2	0.907	0.898	0.906	0.764	0.891	0.890	0.888
p=2.5	0.770	0.765	0.771	0.875	0.899	0.757	0.757
p→3	0.673	0.669	0.674	0.883	0.894	0.660	0.658
FAMWK-Means							
p→1	0.910	0.904	0.907	0.807	0.894	0.893	0.897
p=1.1	0.913	0.907	0.913	0.811	0.895	0.895	0.897
p=1.2	0.916	0.907	0.918	0.816	0.897	0.900	0.901
p=1.3	0.917	0.907	0.916	0.813	0.899	0.896	0.898
p=1.4	0.915	0.910	0.918	0.847	0.905	0.897	0.900
p=1.5	0.918	0.908	0.920	0.830	0.905	0.897	0.901
p=1.6	0.920	0.914	0.920	0.822	0.908	0.901	0.905
p=1.7	0.924	0.914	0.923	0.766	0.907	0.899	0.903
p=1.8	0.924	0.913	0.923	0.770	0.912	0.903	0.901
p=1.9	0.927	0.918	0.925	0.775	0.916	0.905	0.905
p=2	0.929	0.919	0.924	0.784	0.915	0.910	0.909
p=2.5	0.792	0.784	0.792	0.896	0.915	0.773	0.778
p→3	0.696	0.690	0.693	0.899	0.914	0.679	0.681
FAMWK-Means with Best Feature Adjustment							
p→1	0.931	0.928	0.931	0.824	0.912	0.912	0.914
p=1.1	0.933	0.926	0.932	0.829	0.916	0.915	0.918
p=1.2	0.935	0.927	0.934	0.836	0.917	0.921	0.922
p=1.3	0.935	0.931	0.935	0.835	0.919	0.918	0.922
p=1.4	0.938	0.927	0.937	0.869	0.921	0.919	0.921
p=1.5	0.940	0.930	0.941	0.847	0.925	0.917	0.920
p=1.6	0.941	0.933	0.940	0.843	0.928	0.923	0.924

p=1.7	0.945	0.931	0.943	0.786	0.926	0.921	0.921
p=1.8	0.943	0.933	0.944	0.790	0.931	0.921	0.922
p=1.9	0.944	0.936	0.943	0.794	0.933	0.925	0.926
p=2	0.948	0.938	0.946	0.804	0.933	0.929	0.930
p=2.5	0.811	0.808	0.809	0.914	0.939	0.797	0.796
p→3	0.716	0.707	0.712	0.919	0.938	0.698	0.699

Table 5. Rand Index for Different Methods Applied in Data Sets with 50% Noise

	Silhouette						
	Eucl	Manh	Mink	DB	Wint	HS	Dunn
K-Means	0.775	0.679	0.768	0.786	0.811	0.729	0.821
			MWK- Means				
p→1	0.850	0.840	0.845	0.741	0.829	0.830	0.829
p=1.1	0.849	0.842	0.848	0.748	0.833	0.830	0.832
p=1.2	0.850	0.842	0.857	0.749	0.831	0.837	0.843
p=1.3	0.857	0.842	0.854	0.754	0.834	0.835	0.839
p=1.4	0.852	0.844	0.856	0.790	0.844	0.840	0.838
p=1.5	0.861	0.845	0.854	0.762	0.841	0.835	0.839
p=1.6	0.852	0.847	0.854	0.759	0.840	0.841	0.839
p=1.7	0.862	0.847	0.863	0.704	0.851	0.842	0.843
p=1.8	0.866	0.848	0.860	0.703	0.846	0.843	0.836
p=1.9	0.860	0.853	0.866	0.719	0.848	0.839	0.846
p=2	0.866	0.860	0.863	0.721	0.853	0.849	0.853
p=2.5	0.726	0.721	0.730	0.829	0.857	0.708	0.711
p→3	0.635	0.625	0.626	0.842	0.854	0.614	0.613
			FAMWK-Means				
p→1	0.919	0.912	0.917	0.813	0.901	0.899	0.902
p=1.1	0.922	0.913	0.923	0.822	0.907	0.909	0.903
p=1.2	0.923	0.914	0.922	0.828	0.903	0.911	0.906
p=1.3	0.925	0.920	0.925	0.824	0.911	0.904	0.906
p=1.4	0.929	0.920	0.926	0.854	0.909	0.906	0.910
p=1.5	0.924	0.915	0.926	0.835	0.914	0.904	0.909
p=1.6	0.928	0.920	0.926	0.830	0.915	0.911	0.914
p=1.7	0.934	0.922	0.930	0.778	0.914	0.910	0.916
p=1.8	0.935	0.924	0.936	0.776	0.922	0.912	0.913
p=1.9	0.937	0.925	0.937	0.787	0.920	0.915	0.913
p=2	0.938	0.931	0.937	0.788	0.925	0.917	0.921
p=2.5	0.795	0.795	0.796	0.908	0.923	0.781	0.787
p→3	0.699	0.698	0.703	0.911	0.925	0.684	0.690
			FAMWK-Means with Best Feature Adjustment				
p→1	0.907	0.903	0.906	0.803	0.888	0.891	0.895
p=1.1	0.913	0.904	0.913	0.807	0.896	0.893	0.899
p=1.2	0.911	0.906	0.911	0.818	0.892	0.896	0.899
p=1.3	0.917	0.905	0.919	0.813	0.897	0.895	0.895
p=1.4	0.914	0.905	0.916	0.848	0.899	0.898	0.899
p=1.5	0.915	0.904	0.921	0.826	0.905	0.899	0.896
p=1.6	0.920	0.909	0.917	0.819	0.902	0.903	0.902
p=1.7	0.924	0.914	0.919	0.766	0.906	0.900	0.900
p=1.8	0.923	0.908	0.924	0.769	0.914	0.902	0.898
p=1.9	0.922	0.912	0.924	0.773	0.914	0.902	0.907
p=2	0.929	0.922	0.927	0.777	0.915	0.907	0.908
p=2.5	0.789	0.787	0.792	0.895	0.912	0.778	0.775
p→3	0.692	0.691	0.690	0.900	0.912	0.675	0.676

In the second group of experiments, 50% noise feature was added to each of our data sets. In this case, we find that the *K*-means method is less effective than the previous one under various of validity index. This indicates that noise has a great impact on the *K*-means algorithm. At the same time, the results produced by MWK-Means algorithm is not as good as the previous one. That shows MWK-Means is not able to find the true structure of the data set. We can see that our method produces better results.

Table 4 shows FAMWK-Means has achieved its maximum value of 0. 929 by Silhouette using the Euclidean distance. Compared to MWK-Means, FAMWK-Means significantly improved the ability of

clustering, on average, higher than 0.02. FAMWK-Means with Best Feature Adjustment produce the best result of this experiment at p=2 using Silhouette index.

Table 5 shows that FAMWK-Means presents the highest rand index of 0.938 at p=2 using silhouette index using Euclidean distance. Compared to the first set of experiment, results provided by MWK-MEANS are clearly worse. In the no noise data sets, the best value of the Rand index provided by MWK-MEANS is reached 0.961 at p=2, while in the 50 noise data sets, its best value drops to 0.866 at p=2. On the contrary, although FAMWK-Means does not provide the better results in the noise data sets than in the no noise data sets, FAMWK-Means improves the results dramatically compared to the MWK-MEANS. This improvement is 0.08.

In the third group, we add 100% noise features to each data set. Table 6 and 7 show the results for this set of experiments regarding F-measure and the rand index.

Table 6. F-measure of Different Methods Applied in Data Sets with 100% Noise

	Silhouette						
	Eucl	Manh	Mink	DB	Wint	HS	Dunn
K-Means	0.617	0.546	0.618	0.683	0.733	0.687	0.782
			MWK- Means				
p→1	0.869	0.867	0.870	0.763	0.851	0.851	0.854
p=1.1	0.872	0.864	0.873	0.767	0.856	0.852	0.854
p=1.2	0.874	0.867	0.874	0.774	0.854	0.857	0.861
p=1.3	0.874	0.870	0.876	0.778	0.859	0.853	0.857
p=1.4	0.879	0.870	0.876	0.806	0.862	0.857	0.859
p=1.5	0.877	0.868	0.877	0.791	0.864	0.858	0.860
p=1.6	0.880	0.870	0.881	0.785	0.865	0.864	0.866
p=1.7	0.882	0.871	0.879	0.727	0.870	0.861	0.864
p=1.8	0.883	0.872	0.884	0.730	0.871	0.863	0.863
p=1.9	0.881	0.876	0.888	0.739	0.871	0.866	0.863
p=2	0.886	0.878	0.884	0.744	0.870	0.869	0.868
p=2.5	0.748	0.744	0.750	0.855	0.877	0.736	0.735
p→3	0.652	0.649	0.654	0.863	0.873	0.639	0.638
			FAMWK- Means				
p→1	0.889	0.882	0.885	0.787	0.874	0.872	0.876
p=1.1	0.892	0.887	0.893	0.790	0.874	0.874	0.875
p=1.2	0.894	0.886	0.897	0.794	0.875	0.878	0.881
p=1.3	0.896	0.886	0.895	0.792	0.879	0.876	0.876
p=1.4	0.894	0.889	0.897	0.826	0.885	0.876	0.879
p=1.5	0.897	0.888	0.900	0.808	0.885	0.876	0.880
p=1.6	0.899	0.893	0.898	0.801	0.887	0.880	0.884
p=1.7	0.902	0.894	0.902	0.745	0.885	0.879	0.883
p=1.8	0.904	0.892	0.903	0.749	0.890	0.882	0.880
p=1.9	0.906	0.897	0.904	0.753	0.894	0.885	0.883
p=2	0.908	0.897	0.903	0.763	0.893	0.889	0.888
p=2.5	0.771	0.764	0.770	0.876	0.893	0.752	0.758
p→3	0.676	0.669	0.673	0.879	0.893	0.658	0.659
			FAMWK-Means with Best Feature Adjustment				
p→1	0.910	0.906	0.910	0.804	0.892	0.890	0.893
p=1.1	0.912	0.905	0.912	0.808	0.895	0.893	0.897
p=1.2	0.915	0.906	0.914	0.814	0.895	0.900	0.901
p=1.3	0.914	0.910	0.913	0.815	0.899	0.897	0.901
p=1.4	0.917	0.905	0.916	0.848	0.900	0.898	0.901
p=1.5	0.918	0.909	0.920	0.827	0.904	0.896	0.900
p=1.6	0.921	0.911	0.920	0.822	0.907	0.902	0.904
p=1.7	0.924	0.910	0.922	0.765	0.904	0.900	0.901
p=1.8	0.922	0.912	0.924	0.770	0.910	0.900	0.902
p=1.9	0.922	0.915	0.921	0.774	0.913	0.903	0.905
p=2	0.927	0.916	0.925	0.784	0.913	0.908	0.909
p=2.5	0.790	0.787	0.787	0.893	0.917	0.777	0.775
p→3	0.695	0.685	0.690	0.898	0.917	0.676	0.677

Table 7 shows that all the cluster validity indices are improved by using FAMWK-Means with Best Feature, at various values of *p*. The best result overall was obtained by FAMWK-Means with Best Feature Adjustment using the distance at *p*=1.8.

Table 7. Rand Index for Different Methods Applied in Data Sets with 100% Noise

	Silhouette						
	Eucl	Manh	Mink	DB	Wint	HS	Dunn
K-Means	0.725	0.639	0.728	0.666	0.711	0.659	0.726
			MWK- Means				
p→1	0.820	0.806	0.813	0.709	0.798	0.798	0.797
p=1.1	0.816	0.812	0.816	0.717	0.801	0.800	0.799
p=1.2	0.817	0.810	0.827	0.715	0.800	0.807	0.811
p=1.3	0.824	0.808	0.821	0.720	0.802	0.800	0.807
p=1.4	0.818	0.812	0.826	0.758	0.811	0.807	0.805
p=1.5	0.827	0.812	0.822	0.728	0.807	0.804	0.809
p=1.6	0.819	0.816	0.824	0.725	0.806	0.806	0.804
p=1.7	0.830	0.814	0.831	0.673	0.819	0.810	0.811
p=1.8	0.836	0.817	0.828	0.671	0.813	0.813	0.805
p=1.9	0.829	0.819	0.834	0.688	0.817	0.804	0.813
p=2	0.831	0.828	0.829	0.689	0.821	0.814	0.821
p=2.5	0.694	0.688	0.699	0.795	0.827	0.675	0.679
p→3	0.601	0.595	0.596	0.808	0.820	0.580	0.581
			FAMWK-Means				
p→1	0.885	0.880	0.886	0.781	0.870	0.866	0.871
p=1.1	0.891	0.880	0.889	0.791	0.872	0.875	0.868
p=1.2	0.891	0.881	0.889	0.798	0.870	0.878	0.873
p=1.3	0.891	0.885	0.892	0.791	0.880	0.872	0.874
p=1.4	0.898	0.889	0.891	0.821	0.875	0.875	0.876
p=1.5	0.891	0.884	0.895	0.801	0.880	0.871	0.878
p=1.6	0.897	0.887	0.893	0.798	0.881	0.877	0.882
p=1.7	0.900	0.888	0.897	0.748	0.883	0.878	0.882
p=1.8	0.904	0.890	0.902	0.745	0.889	0.880	0.880
p=1.9	0.899	0.894	0.904	0.753	0.888	0.882	0.882
p=2	0.899	0.900	0.901	0.758	0.891	0.886	0.889
p=2.5	0.761	0.761	0.764	0.874	0.888	0.746	0.754
p→3	0.667	0.664	0.668	0.879	0.892	0.653	0.658
			FAMWK-Means with Best Feature Adjustment				
p→1	0.912	0.907	0.912	0.806	0.890	0.893	0.900
p=1.1	0.917	0.905	0.918	0.808	0.901	0.894	0.902
p=1.2	0.915	0.910	0.917	0.823	0.894	0.897	0.903
p=1.3	0.922	0.907	0.920	0.815	0.903	0.897	0.899
p=1.4	0.920	0.910	0.922	0.853	0.905	0.901	0.901
p=1.5	0.919	0.908	0.922	0.829	0.908	0.901	0.901
p=1.6	0.925	0.910	0.922	0.822	0.905	0.908	0.905
p=1.7	0.929	0.915	0.924	0.768	0.911	0.904	0.901
p=1.8	0.927	0.912	0.929	0.773	0.919	0.905	0.904
p=1.9	0.925	0.916	0.926	0.777	0.919	0.907	0.909
p=2	0.928	0.927	0.928	0.782	0.917	0.912	0.912
p=2.5	0.793	0.791	0.795	0.896	0.914	0.781	0.779
p→3	0.697	0.693	0.692	0.902	0.915	0.681	0.679

In order to illustrate the results of the four methods more clearly, Fig. 1 shows the relationship between the F-measure value and the *P* value in the absence of the noise feature. Here, we adopt the Silhouette index (using Minkowski distance). Fig. 2 shows the relationship between the F-measure value and the P value with 50% extra noise feature using the Silhouette index (using Minkowski distance). Fig. 3 shows the relationship between the F-measure value and the *P* value with the 100% extra noise feature using the Silhouette index (using Minkowski distance).

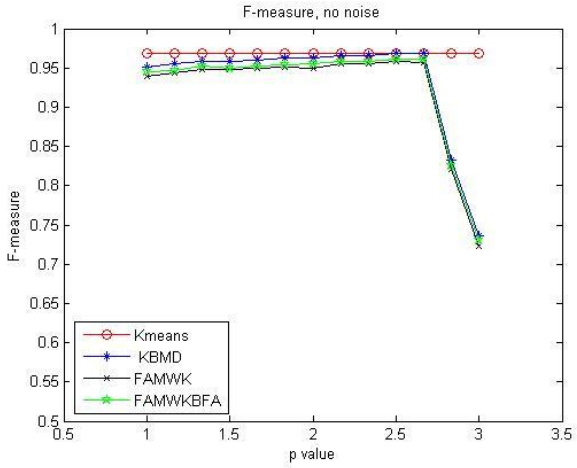

Fig. 1. F-measure of different method on no noise data sets.

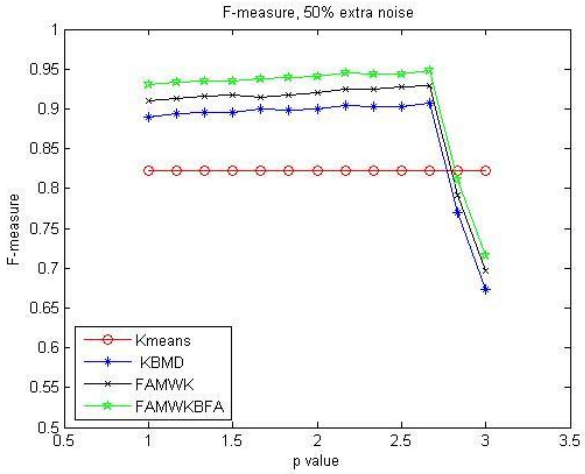

Fig. 2. F-measure of different method on 50% extra noise data sets.

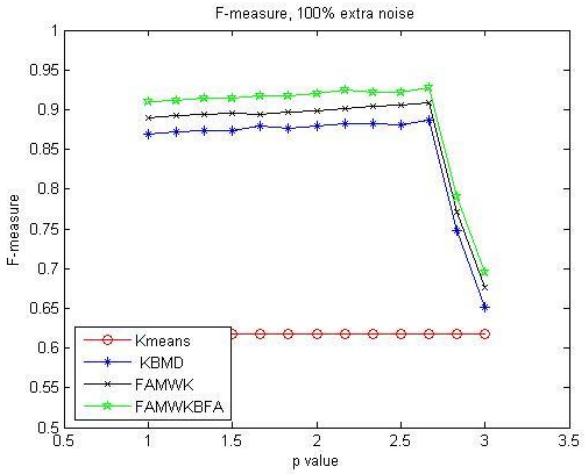

Fig. 3. F-measure of different method on 100% extra noise data sets.

From the tables, it can be inferred that in the case of noise features, the effect has been significantly improved by the methods we proposed.

4.3. Time Consumption Analysis

K-Means, MWK-Means, FAMWK-Means are carried on the 12 different data sets 10 times respectively, and

the average time is obtained as the final result..

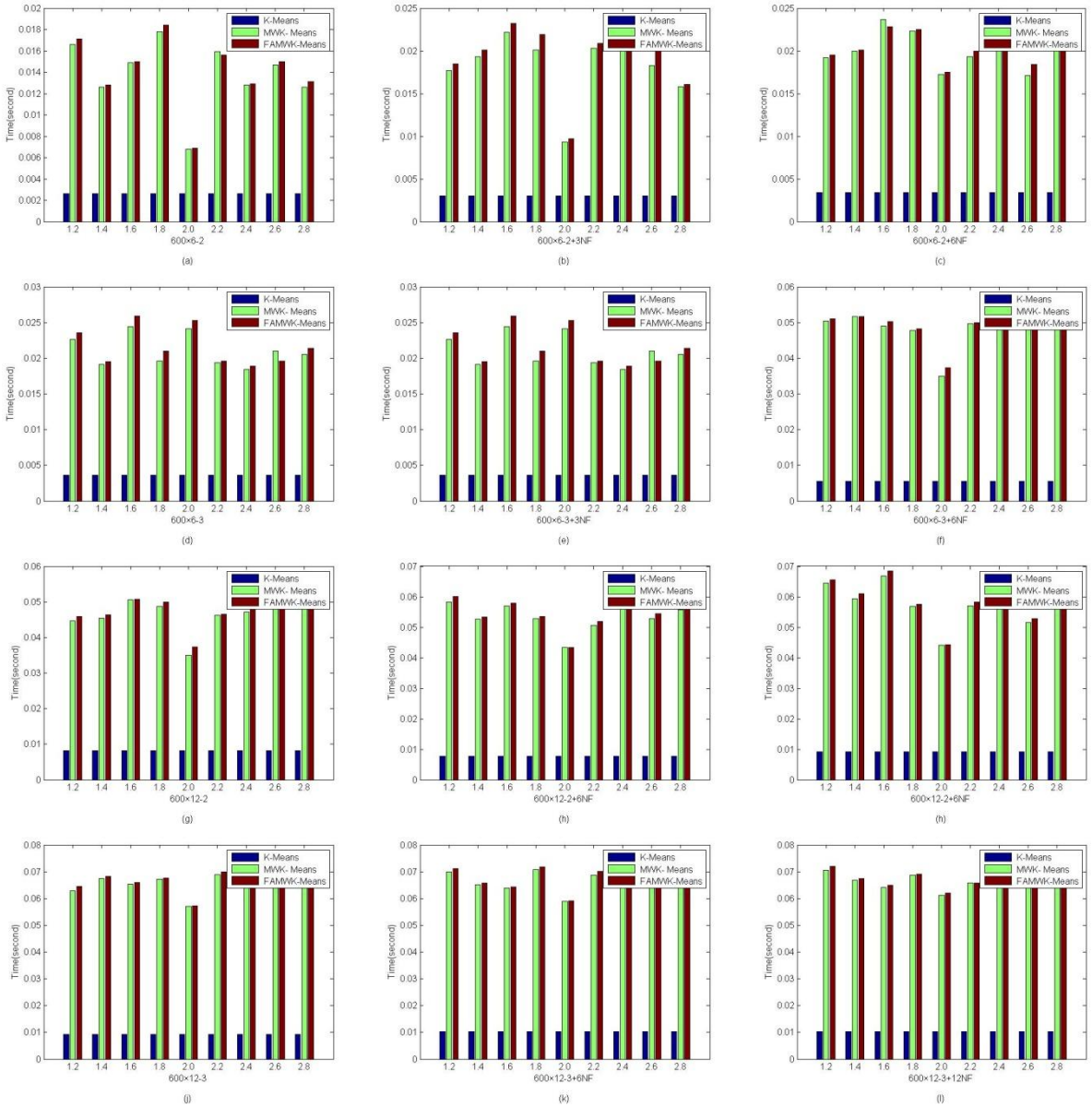

Fig. 4. F- The Average Time and the different *P* value over 10 runs on the 12 data sets.

From Fig. 4, we can see that the time consumption of K-Means is lowest on every data set. When the entities and features of the data sets increase, the time consumptions of the three algorithms raise accordingly, but not significantly. Although MWK-Means spends less time than FAMWK-Means in most of cases, they have similar values. In the most time consuming case, our algorithm has no more than 1 second.

5. Conclusion

This paper proposes three methods aiming at improving the precision to retrieve the true cluster number of spherical Gaussian data sets. These methods can describe the structure of data sets with or without noise features in different degrees, reduce the number of iterations, and improve the stability of the clustering algorithm. Feature weight is introduced to the similarity measure. According to the results of each iteration, calculate the ratio of distance between clusters and within clusters on each of the features, execute a certain degree of reduction in Minkowski space and eliminate the clustering effect of irrelevant features, so as to

reflect the similarity degree between objects more accurately. Experimental results show that compared with the traditional *K*-means algorithm, the clustering accuracy rate is higher, the clustering results are more stable, the algorithm can effectively improve the imbalanced medical data sets clustering performance.

Acknowledgment

This work was supported in part by a grant from the National Basic Research Program of China (No.2012CB720702).

References

[1] Gibert, K., Marre, M., & Codina, V. (2010). Choosing the right data mining technique: classification of methods and intelligent recommendation. *International Environmental Modeling and Software Society.*

[2] Liu, P. Y., Zhu, Z. F., & Zhao, L. (2009). Research on information retrieval system based on ant clustering algorithm. *Journal of Software*, 1032-1036.

[3] Cades, I., Smyth, P., & Mannila, H. (2001). Probabilistic modeling of transactional data with applications to profiling, visualization and prediction. *Proceedings of the 7th ACM SIGKDD.* San Francisco: ACM Press.

[4] Li, F., & Zhu, Q. X. (2011). Document clustering in research literature based on NMF and testor theory. *Journal of Software*, 78-82..

[5] Jain, A. K., & Dubes, R. C. (2011). Algorithms for clustering data. *Prentice-Hall Advanced Reference Series*, 1-334.

[6] Lee, G. (2015). Hierarchical clustering using one-class support vector machines. *Symmetry*, 1164–1175.

[7] Obulkasim A., Meijer, A. G., & Van, D. W. M. (2015). Semi-supervised adaptive-height snipping of the hierarchical clustering tree. *BMC Bioinformatics* , 1–11.

[8] Oviedo, B., Serafín, M., & Amilkar, P. (2016). A hierarchical clustering method: Applications to educational data. *Intelligent Data Analysis*, 933-951.

[9] Zhang B, Srihari SN. Properties of binary vector dissimilarity measures. *Proceedings of the JCIS Cvprip.*

[10] Kumar, P., Krishna, P. R., Bapi, R. S., & De, S. K. (2007). Rough clustering of sequential data. *Data and Knowledge Engineering, 3(2)*, 183-199.

[11] Huang, L., Zhao, S. Q., & Wu, X. (2016). Feature selection based on partition clustering. *International Journal of Knowledge-based and Intelligent Engineering Systems*, 135-142.

[12] Tiwari, P., Subhojit, G., Rakesh, K. S. (2015). Classification of two class motor imagery tasks using hybrid GA-PSO based k-means clustering. *Computational Intelligence and Neuroscience*.

[13] Moftah, H. M., *et al*. (2014). Adaptive k-means clustering algorithm for MR breast image segmentation. *Neural Computing and Applications,* 1917-1928.

[14] Chaturvedi, A. D., Green, P. E., & Carroll, J. D. (2001). K-modes clustering. *Journal of Classification*.

[15] Goodman, L. A. (1974). Exploratory latent structure analysis using both identifiable and unidentifiable models. *Biometrika*.

[16] Huang, Z. X., & Michael, K. (2003). A note on K-modes clustering. *Journal of Classification*, 257-26.

[17] Sun, Y., Zhu, Q. M., & Chen, Z. X. (2002). An iterative initial-points refinement algorithm for categorical data clustering. *Pattern Recognition Letters*, 875-884.

[18] Belhaouari, S. B., Shahnawaz, A., & Samer, M. (2014). Optimized K-means algorithm. *Mathematical Problems in Engineering.*

[19] Liao, H. H., *et al.* (2014). Adaptive initialization method based on spatial local information for-means algorithm. *Mathematical Problems in Engineering*.

[20] Ding, C., & He, X. (2004). K-Nearest-Neighbor in data clustering: Incorporating local information into global optimization. *Proceedings of the ACM Syrup. On Applied Computing*.

[21] Peralta, B., Pablo, E., & Alvaro, S. (2013). Enhancing K-Means using class labels. *Intelligent Data Analysis*, 1023-1039.

[22] Yang, M. S., Hu, Y. J., Lin, K. C. R., & Lin, C. C. L. (2002). Segmentation techniques for tissue differentiation in MRI of ophthalmology using fuzzy clustering algorithm. *Journal of Magnetic Resonance Imaging, (20)*, 173-179.

[23] Li, J., Gao, X. B., & Jiao, L. C. (2006). A new feature weighted fuzzy clustering algorithm. *ACTA Electronical Sinica, 34(1)*, 4l2-420.

[24] Yang, T., & Jun, W. (2014). A robust K-Means type algorithm for soft subspace clustering and its application to text clustering. *Journal of Software* , 2120-2124.

[25] Cai, W. L., Chen, S. C., & Zhang, D. Q. (2007). Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation. *Pattern Recognition, 40(3)*, 825-833.

[26] Harel, D., & Koren, Y. (2001). Clustering spatial data using random walks. *Proceedings of the 7th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*..

[27] Saha, A., & Swagatam, D. (2015). Categorical fuzzy k-modes clustering with automated feature weight learning. *Neurocomputing* , 422-435.

[28] Estivill-Castro, V., & Lee, I. (2000). Autoclust: Automatic clustering via boundary extraction for mining massive point-data sets. *Proceedings of the 5th Int'l Conf. on Geocomputation*.

[29] Khan, S. S., & Amir, A. (2013). Cluster center initialization algorithm for K-modes clustering. *Expert Systems with Applications* , 7444-7456.

[30] Zhao, Y. C., & Song, J. (2001). GDILC: A grid-based density isoline clustering algorithm *Proceedings of the Internet Conf. on Info-Net*.

[31] Ma, W. M., Chow, E., & Tommy, W. S. (2004). A new shifting grid clustering algorithm. *Pattern Recognition, 37(3)*, 503-514.

[32] Pilevar, A. H., & Sukumar, M. GCHL: A grid-clustering algorithm for high-dimensional very large spatial databases. *Pattern Recognition*.

[33] Nanni, M., & Pedreschi, D. (2006). Time-focused clustering of trajectories of moving objects. *Journal of Intelligent Information Systems, 27(3)*, 267-289.

[34] Birant, D., & Kut, A. (2007). ST-DBSCAN: An algorithm for clustering spatial-temporal data. *Data and Knowledge Engineering, 60(1)*, 208-221.

[35] Sergio, J., Fabio, A. G., & Alexander, G. (2016). Mathematical properties of soft cardinality: Enhancing Jaccard, Dice and cosine similarity measures with element-wise distance. *Information Sciences,* 373-389.

[36] Flury, B. K., & Riedwyl, H. (1986). Standard distance in univariate and multivariate analysis. *The American Statistician, 40,* 249–251.

[37] Gouda, K., & Arafa, M. (2015). An improved global lower bound for graph edit similarity search. *Pattern Recognition Letters*, 8–14.

[38] Peter, J. R. Silhouettes: a Graphical aid to the interpretation and validation of cluster analysis. *Computational and Applied Mathematics,* 53–65.

[39] Davies, D. L., & Bouldin, D. W. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 224–227.

[40] Renato, C. D. A. (2012). Constrained clustering with Minkowski weighted K-Means. *Proceedings of the 13th IEEE International Symposium on Computational Intelligence and Informatics*.

[41] Renato, C. D. A., & Peter K. (2012). On initializations for the minkowski weighted k-means. *Advances in*

Intelligent Data Analysis XI Lecture Notes in Computer Science.

[42] Peiravi, A., & Kheibari, H. T. (2008). A fast algorithm for connectivity graph approximation using modified Manhattan distance in dynamic networks. *Applied Mathematics and Computation.*

[43] Ghosh, A., & Barman, S. (2016). Application of Euclidean distance measurement and principal component analysis for gene identification. *Gene*.

[44] Mousa, A., & Yusof, Y. (1691). An improved chebyshev distance metric for clustering medical images. *AIP Conference Proceedings*.

Qiang Zhan was born in Yuncheng, Shanxi, China. He received his M.S. degree from Taiyuan University of Technology, China in 2005. Now he is studying for the Ph.D. degree in the School of Computer Science and Technology in Beijing Institute of Technology. His research interests are cloud computing, cluster analysis and big data analytics.