A K-means Algorithm Based On Feature Weighting

Yan Xu[1], Xueliang Fu[1,*] , Honghui Li[1] , Gaifang Dong[1] and Qing Wang[1]

[1]College of Computer and Information Engineering, Inner Mongolia Agricultural University, Hohhot, Inner Mongolia 010020, China;

Abstract. Cluster analysis is a statistical analysis technique that divides the research objects into relatively homogeneous groups. The core of cluster analysis is to find useful clusters of objects.K-means clustering algorithm has been receiving much attention from scholars because of its excellent speed and good scalability.However, the traditional K-means algorithm does not consider the influence of each attribute on the final clustering result, which makes the accuracy of clustering have a certain impact.In response to the above problems, this paper proposes an improved feature weighting algorithm.The improved algorithm uses the information gain and ReliefF feature selection algorithm to weight the features and correct the distance function between clustering objects, so that the algorithm can achieve more accurate and efficient clustering effect.The simulation results show that compared with the traditional K-means algorithm, the improved algorithm clustering results are stable, and the accuracy of clustering is significantly improved. .

1 Introduction

Data mining is currently a hot topic in the field of artificial intelligence and database research. It refers to the process of extracting implicit information and knowledge that are hidden in advance but are potentially useful information from a large amount of data. Cluster analysis has become a very important research direction in data mining. The K-means algorithm proposed by McQueen[1] is one of the most commonly used methods in cluster analysis. It uses distance as an evaluation index for similarity, that is, the closer the distance between two objects is, the greater the similarity. The algorithm considers the cluster is composed of objects that are close together, so the compact and independent cluster is the ultimate target[2]. The K-means algorithm assumes that each feature of the sample contributes the same degree to the final cluster.In the actual situation, some features play a big role in the clustering process or the effects of some features are small and have no effect on the clustering process.

For the problem of traditional K-means algorithm, scholars have done a large number of studies and studies have shown that by assigning different feature weights to the features, the above problems can be effectively solved and the clustering performance can be improved. Currently, there are many algorithms for calculating feature weights: Liu Ming[3] et al. proposed a feature weight quantization function combined with restricted data. This function quantifies feature weights by user-specified restriction data and assigns different confidence levels to different restricted data, which solves the problem of limiting data distribution unevenness and restricting data inclusion inconsistency. LiJie[4] et al. proposed to apply the ReliefF algorithm for classification problems to clustering problems, calculate feature weight values by ReliefF algorithm, and weight each dimension feature to improve cluster performance. Meng Qian[5] et al. proposed that to assign weights to each feature and weight them by using the gradient descent technique to minimize the feature evaluation function FLearning(w). The algorithm uses the advantages of genetic algorithm and simulated annealing algorithm to weaken the influence of redundant features and solve the problem of easily falling into local optimal solution. Songtao Shang[6] et al. proposed an improved Gini index algorithm to calculate feature weights. This algorithm overcomes the shortcomings of the original Gini, combines the conditional probability with the posterior probability, and suppresses the influence of the training set imbalance. Ouyang Hao [7] used the information gain in information theory to calculate feature weights and weight each feature, effectively solving the influence of features on clustering.

In summary, in order to improve the clustering accuracy of the traditional K-means algorithm, scholars at home and abroad have carried out a lot of improvement research on the K-means algorithm, and achieved some phased results. This paper intends to study the contribution of each feature of the clustering process to the clustering result in the traditional K-means algorithm, so that the features with large contribution degree are used preferentially. In theory, the accuracy rate and precision of K-means algorithm clustering can be effectively improved. Therefore, this paper proposes an organic fusion of information gain and ReliefF feature selection algorithm. By using information gain and ReliefF feature selection algorithm to weight the feature, the distance function between cluster objects is corrected, and the algorithm achieves more accurate and efficient clustering effect. The experimental results show that the improved algorithm clustering results are stable and have high accuracy and achieve the intended purpose.

* Corresponding author: fuxl@imau.edu.cn

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (http://creativecommons.org/licenses/by/4.0/).

2 K-means algorithm

The core idea of the K-means algorithm is to iteratively divide the data objects into different clusters so as to minimize the objective function so that the generated clusters are as compact and independent as possible. The specific flow of the algorithm is as follows.

Input: Number k of clusters, data set D containing n objects.

Output: k clusters.

Proceed as follows:

(1) Select k objects arbitrarily from D as the initial clustering center;

(2) Calculate the distance between each object and these central objects; and repartition the corresponding objects according to the minimum distance;

(3) Recalculate the mean of each cluster;

(4) When certain conditions are satisfied, E.g no objects are re-assigned to other clusters, the cluster center no longer changes, and the sum of squared errors (SSE) is minimal, the algorithm terminates; if the conditions are not met, go back to step (2).

The distance between each object and the center object is Euclidean distance. The distance formula is as follows:

$$d(x, y) = \sqrt{\sum_{j=1}^{m} (x_j - y_j)^2} \qquad (1)$$

In the above formula: x, y represent the sample and cluster center respectively; j represents the j-th dimension feature.

3 Improved algorithm based on feature weighting

3.1. Information Gain

The information gain indicates the degree to which the uncertainty of the information is reduced, that is the amount of change in information entropy before and after classification.

Information entropy represents the uncertainty of information, and its mathematical expression is as follows:

$$H(x) = -\sum_{i=1}^{m} p_i \log_2 p_i \qquad (2)$$

where p_i indicates the probability of an event occurring.

Let data set X, feature set $A=\{A_1,A_2,.....,A_m\}$, data set X is divided into n parts $X=\{x_1,x_2,.....,x_n\}$according to feature A_j.The expected entropy of characteristic A_j to data set X is H ($X|A_j$)and the formula is as follows:

$$H(X \mid A_j) = \sum_{i=1}^{n} \frac{|x_i|}{|X|} H(x_i) \qquad (3)$$

The calculation formula of the information gain Gain(X, A_j) of the feature A_j for the data set X is as follows:

$$Gain(X, A_j) = H(X) - H(X \mid A_j) \qquad (4)$$

The information gain represents the difference in information uncertainty before and after classification. In the clustering process, if the information gain value is larger, the contribution of the feature to the clustering result is greater.

3.2. RliefF algorithm

In 1994, Knonenko proposed the ReliefF algorithm, which is an extension of the Relief algorithm and deals with multiple classification problems[8]. The basic idea of the ReliefF algorithm is to randomly take a sample x_i from the training sample set; then take the k nearest neighbors H_i from the same sample as x_i; then take out k samples M_i from other classes that are different from x_i; the weight of each feature is updated according to the weight formula. The m times are randomly selected to get the final feature weight. The weight expression is as follows:

$$w(j) = w(j) + \sum_{c \neq class(i)} \frac{\frac{p(c)}{1-p(class(x_i))} \sum_{j=1}^{k} d(x_i(j), M_i(j))}{mk} - \sum_{j=1}^{k} \frac{d(x_i(j), H_i(j))}{mk} \qquad (5)$$

In the above formula: $class(i)$ represents the class to which the sample belongs; *c* represents the class other than the class to which the sample belongs; *p(c)* represents the prior probability of the class *c*. $x_i(j)$ represents the value of the sample x_i with respect to the j-th feature; *m* is the number of samples taken randomly; $d\left(x_i(j),\ H_i(j)\right)$shows the distance function, which is used to calculate the distance between the two samples for the j-th feature. Calculated as follows:

$$d(x_i(j), M_i(j)) = \left| \frac{x_i(j) - M_i(j)}{\max(j) - \min(j)} \right| \qquad (6)$$

Among them, $\max(j), \min(j)$ represent the maximum value and the minimum value of all the values of the j-th feature.

The ReliefF algorithm is used to deal with multi-classification problems. Each sample must have an explicit class tag. But there are no class markers in the samples in the cluster analysis. Therefore, we need to perform an initial clustering on the sample set to obtain the class label of the sample, and then use the ReliefF algorithm to calculate the feature weight.

3.3. Improved algorithm GR_Kmeans algorithm based on feature weighting (GainReliefF_Kmeans)

The traditional K-means algorithm assumes that each feature has the same impact on clustering in the clustering process, ignoring the influence of the feature on the clustering process, leading to a lower accuracy of the final clustering result. The improved feature-weighted algorithm effectively solves this problem.

GR_Kmeans algorithm is to cluster the feature weights of the clustering objects and the information gain as the feature weights of the K-means algorithm.Let the information gain weight is w_1 and Feature weight is w_2, The final feature weight is

$$w = \frac{(w_1 + w_2)}{2} \quad (7)$$

The steps of the GR_Kmeans algorithm are as follows:

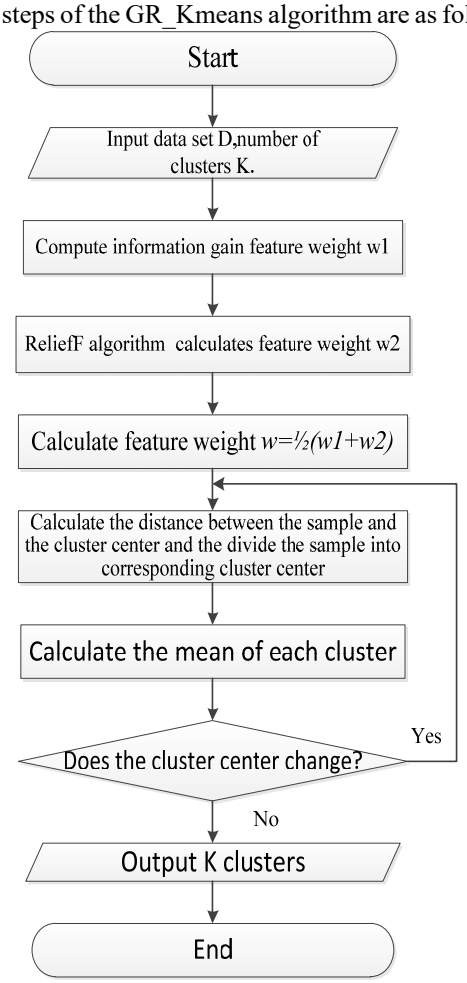

Figure 1. GR_Kmeans algorithm flow chart

Input: data set D, number of clusters K

Output: K clusters

(1) Select K initial cluster centers randomly;

(2) Calculate information gain feature weight w_1;

(3) ReliefF algorithm calculates feature weights w_2;

(4) Calculate feature weights;

(5) Calculate the distance between each sample and these cluster centers $d(x,y)=\sqrt{\sum_{j=1}^{m}\omega_j(x_j-y_j)^2}$; and according to the minimum distance, the samples are divided into corresponding cluster centers;

(6) Recalculate the mean of each cluster;

(7) If the cluster center no longer changes, the algorithm terminates; if the cluster center changes, go back to step (5).

4 Experiment

4.1. Experimental environment and data set

The hardware environment of the experiment is Intel(R)Core(TM)i5-6500 3.20GHz, 8G memory, the software environment is Matlab2016b, Windows7 operating system. The data set selected for the experiment is the Iris, Balance-scale, and Stalog data sets in the UCI[9] database. The main information of the data set is shown in Table 1.

Table 1. Experimental data set

Data Sets	Number of Data	Attributes	Number of Types
Iris	150	4	3
Balance	625	4	3
Stalog	846	18	4

4.2. Experimental Results and Analysis

In order to verify that each feature of the clustering object contributes differently to the clustering result during the clustering process,calculate the weight values corresponding to each feature of the three data sets of Iris, Balance-scale, and Stalog 20 times as shown in Figure 2, Figure 3, Figure 4. One line in the figure represents a calculation

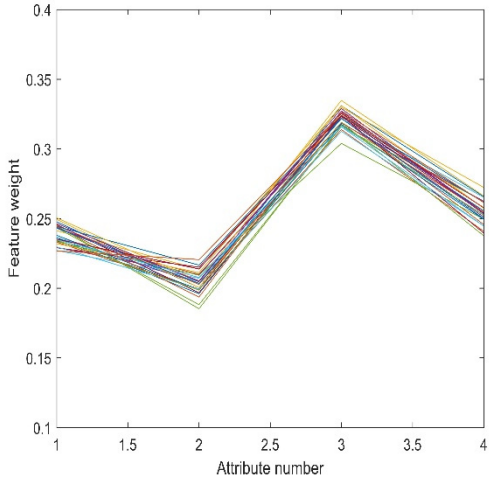

Figure 2. Feature weights of the Iris dataset

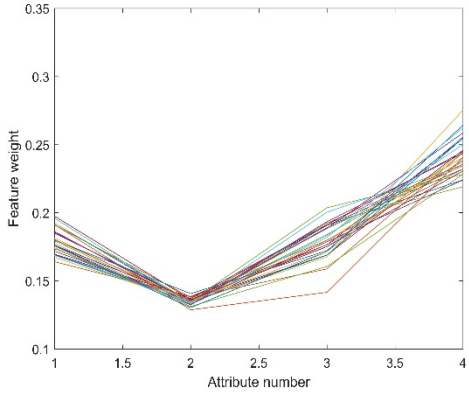

Figure 3. Feature weights for Balance-scale datasets

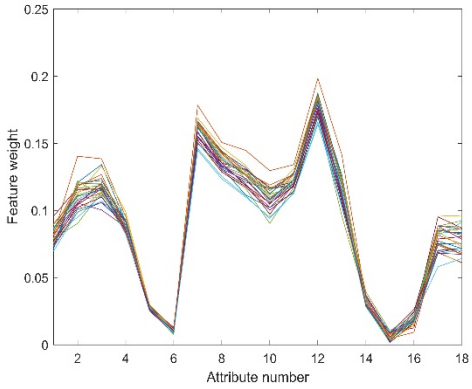

Figure 4. Feature weights of the Stalog data set

From Figure 2, Figure3, and Figure 4, it can be seen that each feature has different effects on the clustering results. Taking the feature weight values of the Stalog dataset in Fig. 4 as an example, it can be seen from the figure that the weight values of feature 7 and feature 12 are relatively high, indicating that they have a greater impact on the clustering results; the weight values of feature 6 and feature 15 are low and almost close to 0, indicating that the impact on the clustering results is small and may not be affected. The traditional K-means algorithm ignores this problem, resulting in a lower final accuracy of the clustering result.

In order to verify the validity and stability of the algorithm, under the same experimental environment, GR_Kmeans algorithm is compared with the traditional K-means algorithm, the weighted K-means algorithm RelieF-kmeans based on RelieF algorithm in literature [4], and the weighted K-means algorithm Gain-kmeans based on information gain.Under the same data set, all algorithms performed 20 separate experiments and averaged values were calculated and compared in terms of accuracy, sum of squared errors (SSE), number of iterations, and runtime. The results are shown in Table 2 - Table 6.

Table 2. Comparison of accuracy of each algorithm in UCI dataset (%)

Data sets	Traditional k-means algorithm	Gain-kmean	RelieF-kmeans[4]	Improved algorithm (GR_Kmeans)
Iris	83.40	87.2	87.07	89.40
Balance	63.03	64.95	64.65	66.09
Stalog	43.97	44.55	44.58	45.32

Table 3. Comparison of error square sum (SSE) of each algorithm in UCI dataset (/ms^2)

Data sets	Traditional k-means algorithm	Gain-kmeans	RelieF-kmeans[4]	Improved algorithm (GR_Kmeans)
Iris	96	82	93	78
Balance	3492	3489	3510	3478
Stalog	3740979	3617939	3562829	3556182

Table 4. Comparison of the number of iterations of each algorithm in the UCI data set (/times)

Data sets	Traditional k-means algorithm	Gain-kmeans	RelieF-kmeans[4]	Improved algorithm (GR_Kmeans)
Iris	8	7	9	9
Balance	14	16	5	9
Stalog	17	16	13	12

Table 5. Runtime comparison of each algorithm in UCI dataset (/ms)

Data sets	Traditional k-means algorithm	Gain-kmeans	RelieF-kmeans[4]	Improved algorithm (GR_Kmeans)
Iris	16.8	6	4.3	4.5
Balance	136.7	36.7	11.4	34.6
Stalog	172.2	52.3	48.4	37.1

Table 6. The average run time of each algorithm in the UCI data set is compared with each iteration (/ms)

Data sets	Traditional k-means algorithm	Gain-kmeans	RelieF-kmeans[4]	Improved algorithm (GR_Kmeans))
Iris	2.1	0.8	0.5	0.5
Balance	9.8	2.3	2.3	3.8
Stalog	10.1	3.2	3.7	3.1

As can be seen from Table 2, in terms of accuracy, GR_Kmeans algorithm is significantly higher than the other three algorithms, because the traditional K-means algorithm ignores the impact of the characteristics of the clustering results, the clustering result is unstable, so accurate the rate is lower than the other three algorithms. As can be seen from Table 3, the sum of squared error of GR_Kmeans algorithm is lower than that of the other three algorithms, and the smaller the squared error is, the more similar the objects in the cluster are, so GR_Kmeans algorithm has a high degree of similarity for each class of objects. The clustering quality is superior to the other three algorithms and achieves the ultimate goal of clustering analysis. That is, the intra-class similarity is high and the similarity between classes is low. As can be seen from Table 4 and Table 5, GR_Kmeans algorithm on Stalod dataset is lower than the other three algorithms in number of iterations and running time. On the Iris dataset, the number of iterations is higher than Gain_kmeans algorithm, but the time is lower than Gain_kmeans algorithm. On the Balance dataset, the number of iterations and the running time are higher than the RelieF-kmeans algorithm. The reason is that the initial clustering center of the algorithm is randomly selected, which leads to the instability of the number of iterations of the algorithm and the length of the running time. But there is little difference between the two on average running time. As can be seen from Table 6, the GR_Kmeans algorithm is lower than the traditional k-means algorithm and higher than RelieF-kmeans algorithm and Gain_kmeans algorithm on Balance dataset in the average run time per iteration. However, the average time of each iteration of GR_Kmeans algorithm is not much different from that of

the two algorithms, indicating that the higher iteration number and running time are affected by the initial clustering center.

5 Conclusion

In this paper, we propose a K-means algorithm GR_Kmeans algorithm based on information gain and ReliefF algorithm for feature weighting, which effectively solves the problem that different features have different effects on clustering. Experimental results show that the improved k-means algorithm is superior to the traditional K-means algorithm and other two feature weighting methods in accuracy and clustering error and good clustering results are obtained.

Acknowledgement

This research was financially supported by Chinese Natural Science Foundations (61363016, 61063004), Key Project of Inner Mongolia Advanced Science Research (NJZZ14100), Inner Mongolia Colleges and Universities Education Department Science Research (NJZC059), Natural Science Foundation of Inner Mongolia Autonomous Region of China (NO.2015MS0605, NO.2015MS0626 NO.2015MS0627, and NO.2017MS0605), and Inner Mongolia Autonomous Region Science and Technology Project (through the precipitation of GSM network on-line monitoring and data transmission system development), and Ministry of Education Scientific research foundation for Study abroad personal[2014] 1685.

References

1. A. Alexandropoulos, F. Plessas, M. Birbas.A dynamic DFI-compatible strobe qualification system for Double Data Rate （ DDR ） physical interfaces.*17th IEEE International Conference on Electronics,Circuits,and Systems （ ICECS ）* (2010)p.277-280.
2. J.W. Zhuo, Y. Zhou. *Quantitative Investment: MATLAB Data Mining Technology and Practice* . Beijing: Publishing House of Electronics Industry, 2017,p.217-224.
3. M. Liu,C. Wu. Similarity calculation based on feature weight evaluation.Chinese Journal of Computers.Vol.**38** (2015)No.07,p. 1420-1433.
4. J. Li, X.B. Gao. A New Feature Weighted Fuzzy Clustering Algorithm. *Proceedings of SPIE - The International Society for Optical Engineering*(2006)p.412-420.
5. Q. Meng. An Improved Clustering Algorithm Based on Feature-weight Learning. Journal of Information and Computational Science.Vol.**12**(2015)No.09,p.3519-3526.
6. S.T. Shang, M.Y. Shi. Improved Feature Weight Algorithm and Its Application to Text Classification. Mathematical Problems in Engineering.Vol.**2016**(2016),p. 1-12.
7. H. Ouyang, Z.W. Wang. Fuzzy Clustering K-prototypes Clustering Algorithm Based on Information Gain. Journal of Computer Engineering and Science,.Vol.**37**(2015)No.05,p. 1009-1014.
8. X.Y. Jian, S.Q.Han. Relief feature selection algorithm on unbalanced data sets. Data Acquisition and Processing.(2016)No.04, p.838-844.
9. UCI Machine Learning Repository: Data Sets. http://archive.ics.uci.edu/ml/datasets.html