A k-means Clustering Algorithm based on Self-Adoptively Selecting Density Radius

***Yang Xinhua*[†], *Yu Kuan*[††], *and Deng Wu*[††]**

[†]School of Mechanical Engineering, Dalian Jiaotong University, Dalian, 116028 China
[††]School of Software, Dalian Jiaotong University, Dalian, 116052 China

Summary
K-means with its rapidity, simplicity and high scalability, has become one of the most widely used text clustering techniques. However, owing to its random selection of initial centers, unstable results were often gotten while using traditional K-means and its variants. Here a new technique of optimizing initial centers of clustering is proposed based on self-adoptively selecting density radius. The result of the experiments shows that K-means with the proposed technique can produce cluster results with high accuracy as well as stability.

Key words:
Text clustering, K-means, Density radius, Self-adoptively

1. Introduction

Along with the popularization of Internet and improvement of enterprise informatization, unstructured text data such as HTML data and free text files or semi-structured text data such as XML data has been increasing at an astonishing speed. Since there is not standard text classification criterion, it is very difficult for people to use the massive text information sources effectively. Therefore, the management and analysis of text data become very important. Nowadays, such fields as text mining, information filtering and information retrieving have brought unprecedented attention to both domestic and foreign experts. As one of the core techniques of text mining, text clustering aims to divide a collection of text documents into different category groups. And the documents in the same category group should be especially similar; the documents in different category group should be of little similarity. This kind of technology can improve the efficiency of information retrieving and utilizing on Internet.

Since 1950s, people have proposed many kinds of clustering algorithms. They may roughly be divided into two kinds, of which one is based on division and the other is based on level. At the same time, a third type, namely the combination of these two methods emerged. Among those based on division clustering algorithms, the most famous is the k- means type algorithm. The basic members of k- means type algorithm family include K-Means, K-Modes [1] and K-Prototypes[2]. K-Means algorithm is used in value data, K-Modes algorithm is used in attribute data, and K-Prototypes algorithm is used in mixed data of value and attribute.

The k- means type algorithm has such advantages as fast speed, easy realization and so on. It is suitable for those kinds of data clustering analysis as in text, picture characteristic and so on. But the iterative process of this algorithm is likely to terminate soon. Therefore, a partially most excellent result can be achieved. Moreover, owing to its random selection of initial centers, unstable results were often gotten. Because clustering is often applied in data which the final user is also unable to judge clustering quality, this kind of unstable results is difficult to accept. Therefore, it is significant to improve the quality and stability of clustering result in text clustering analysis.

2. Traditional K-means Algorithm

2.1 Text Expressed Method Based On Vector Space Model

To apply clustering algorithm in text data, the original text formats have to be transformed into structured forms. The commonly used structured form for text data is Vector Space Model [3]. In this model, text space is regarded as a vector space which is composed by a group of orthogonal term vector. Each text is expressed as a feature vector (namely a line).

Given the text $D_i = (t_{i,1}, w_{i,1}; t_{i,2}, w_{i,2}; \cdots; t_{i,n}, w_{i,n};)$, where $t_{i,j}$ is a feature term; $w_{i,j}$ is the weight of $t_{i,j}$ in the text. The computation of weight is acts according to TF- IDF formula:

$$w_{i,j} = \frac{tf(t_{i,j}, D_j) \times \log\left(N \big/ n_t + 0.01 \right)}{\sqrt{\sum_{j=1}^{m} \left[tf(t_{i,j}, D_j) \times \log\left(N \big/ n_t + 0.01 \right) \right]^2}}$$

Manuscript received August 5, 2006.
Manuscript revised August 25, 2006.

Where $tf\left(t_{i,j}, D_j\right)$ is term frequency of $t_{i,j}$ appearing in D_j, N is the total of text, n_t is the count of text which contain $t_{i,j}$.The weight $w_{i,j}$ has portrayed the ability of term distinguishing text content attribute. The broader a term appears within a document, namely the smaller N / n_t is, the smaller $w_{i,j}$ becomes; thus its ability to distinguishing text attribute is lower, vice versa.

2.2 The K- means Type Algorithm [4]

Let $X = \{X_1, X_2 \cdots\cdots, X_n\}$ be a set of n objects. Object $X_i = \left(x_{i,1}, x_{i,2} \cdots\cdots, x_{i,m}\right)$ is characterized by a set of m variables (attributes). The k-means type algorithms search for a partition of X into k clusters that minimizes the objective function P.

$$P(U,Z) = \sum_{l=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m} u_{i,l}\, d\left(x_{i,j}, z_{l,j}\right)$$

subject to $\sum_{l=1}^{k} u_{i,l} = 1$, $1 \le i \le n$, where:

(i) U is an $n \times k$ partition matrix, $U_{i,l}$ is a binary variable, And $U_{i,l}=1$ indicates that object i is allocated to cluster l;

(ii) $Z = \{Z_1, Z_2 \cdots\cdots Z_k\}$ is a set of k vectors representing the centroids of the k clusters;

(iii) $d(x_{i,j}, z_{l,j})$ is a distance or dissimilarity measure between object i and the centroid of cluster l on the jth variable:

$$d(x_{i,j}, z_{l,j}) = (x_{i,j}, z_{l,j})^2 .$$

The k- means type algorithm process is described as follows:

Input condition: the number of clusters k, as well as the sample collection which contains n data objects;

Output condition: k clusters which Satisfy the variance smallest criterion;

Process flow:

(i) Select k objects randomly from n data objects to take as initial clustering centers;

(ii) Circulate the following step 1 to 2 until no cluster change any longer;

Step1. compute the distances between each object and centroid of its cluster according to average value (central object) of all the objects in a cluster; then divide corresponding object again according to the minimum distance, namely assign the object to the cluster to which central object is the most recent;

Step2. Compute each (has changed) average value (central object) of a cluster again.

3. Self-adoptively Selecting Density Radius to Ascertain Clustering Centers

Initial clustering centers have great impact on k-means type clustering algorithm. And traditionally, they are chosen at random. Therefore, clustering result is usually most superior in part. If they are selected reasonably, clustering result will be more reasonable, moreover clustering speed will also be much faster. In order to make the clustering initial centers dispersive rather than mass, a distance is required. Meanwhile, in order to eliminate the influence from isolated point to clustering result, that is to say isolated point is not considered through algorithm till the end, or it is regarded as another category. Therefore, the concept of density is necessary: Taking an object as the center, and a positive number *r* as the radius, we can get a sphere. The number of other objects which fall in the sphere is called the density of the object. Sort the objects according to the density, and then try to select the objects whose densities are big as initial clustering centers. For those objects whose densities are too small, they can be regarded as isolated point. The method is as follows[5]:

First, set two positive numbers *r* and *d*. *r* is the radius used to calculate density. *d* is the initial distance between two clustering centers. Generally *r* should be less than *d*.

Then, calculate each object's density taking *r* as the radius; sort the objects according to the density. Select the object of the biggest density as the first clustering center. Afterwards, calculate the distance between the first center and the object of which density is the second. If the distance is smaller than *d*, then leave this spot out, otherwise select it as the 2nd center. Then pick out the other object, calculate its distance with the first two clustering centers. If it is smaller than *d*, then leave out, otherwise select it as the 3rd center. Determine other centers by using this method.

The initial clustering centers so selected are at a rather long distance with each other, therefore avoid being too close or centralized and affecting clustering result. In addition, the order of objects' initial input have been disrupted after this process, which makes it possible to input object according to density size. So the algorithm is not sensitive to the input order, accordingly better clustering results will be obtained.

But there may be a problem in operation process. Since *r* and *d* are empirical values, it is difficult to know the size of *r* and *d* in advance for the given sample collection. As for *d*, we may assume it a certain multiple of *r*. But as regard to *r*, it is difficult to find the best value for it. If *r* is too big or too small, it will have no significance for object's point density. Thus it will lead to fail to discover the reasonable initial central points. Moreover, *r* is very sensitive. It closely relates with sample data. The number of sample objects, the size of each object's data value, the size of each object's dimension, the value of k, and the object's distribute situation will all greatly effect on appropriate value of *r*. That is to say, for a given sample collection, a certain corresponding appropriate *r* value should be set.

Therefore, this paper proposes a method on self-adoptively selecting best density radius. It is generally expected that the biggest point density should be equivalent to or smaller than the count of objects in one cluster. So we consider dividing n(the number of all sample objects) by *k* to obtain an approximately average object's count of one cluster, then multiply a certain coefficient (for instance 80% and 70%). Thus the greatest density is locked between *80*n*/*k* and *70*n*/*k*. The method is as follows:

First we assign *r* with an initial value. If the largest density in all points is bigger than 80%**n/k,* then *r* subtracts a length of step (for instance 0.01). In sequence we compute the largest density again. If the largest density is smaller than 70%**n/k*, then *r* adds on a length of step. Then compute the largest density again. Thus the *r* value of the largest density between 80%**n/k* and 70%**n/k* is found. Accordingly the best clustering central points can be further identified. Fig.1 shows the improved clustering algorithm flow.

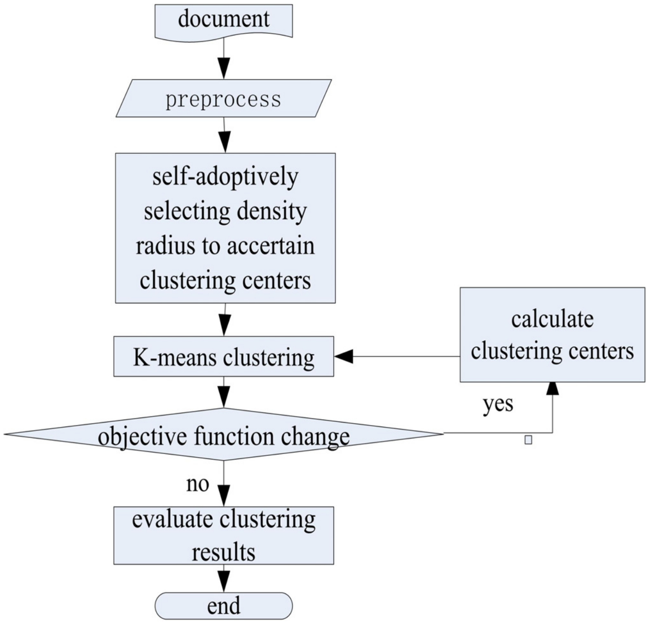
document
preprocess
self-adoptively selecting density radius to accertain clustering centers
K-means clustering
calculate clustering centers
yes
objective function change
no
evaluate clustering results
end

Fig.1 Optimized Algorithm Flow Chart

4. Experimental Result

4.1Experimental Dataset

In the experiment, the dataset derives from the Chinese text classification language materials storehouse -TanCorpV1.0, which is settled by Songbo Tan and Yuefen Wang.
(Http://lcc.software.ict.ac.cn/~tansongbo/corpus1.php).
Some kinds of texts about finance, ball games, campus, movie entertainment, computer science and technology, 20 of each kind are selected from the language materials storehouse. First count the total frequency of each term appearing in all the test dataset and the times of the term appearing in the test texts. Then eliminate the stopped term that has no significance and the high-frequency term (times of its appearance in the test dataset are more than 30% of the test text). Finally select 150 terms whose total frequency is higher than the rest and make them key words. According to TF- IDF formula, calculate the weight of each key word in the corresponding text. Thus create a 100×150 matrix as the initial datum for clustering.

4.2 The Criterion of Algorithm Evaluation

To evaluate the experimental results, this paper employs the commonly used method –Purity— to measure. Suppose that $n_i$is the size of cluster c_i, then the purity definition [6] for the cluster is:

$$S(c_i) = \frac{1}{n_i}\max(n_i')$$

Where n_i' presents the size of intersection between cluster c_i and the *j*th category. So the entire clustering purity definition[6] is:

$$Purity = \sum_{i=1}^{k}\frac{n_i}{n}S(c_i)$$

Where *k* is number of clusters which finally form.

Purity portrays the accuracy of clustering algorithm classification. Generally speaking, the higher is the purity, the more effective clustering algorithm is.

4.3 Experimental Results

To compare validity of algorithm, cluster10 times each by using traditional k- means of algorithm and the optimized initial center k- means of algorithm respectively. Here assign 5 for the k. Each time we disrupt the order of text input randomly before clustering. For the traditional k-means of algorithm, we select k samples as the clustering centers. For the optimized k- means of algorithm, we provide the locked scopes of the biggest density and d values. Tab.1 shows the clustering results of the optimized algorithm.

Coefficient relation between *d* and the *r* vary from sample collection to sample collection .But generally speaking, better clustering effects can only be achieved when r is bigger and d is relatively smaller; or when r is smaller and *d* is relatively bigger. Fig.2 shows the comparison between the two clustering results in purity by using the two algorithms.

Tab. 1 Experimental Datum and Clustering Results

Times	The locked scopes of the biggest density	Value of *d*	Density of center points	Purity
1	80%*n/k--70%*n/k	d=r*1.2	16,16,11,11,8	0.93
2	82%*n/k--72%*n/k	d=r*1.195	16,16,11,10,8	0.89
3	84%*n/k--74%*n/k	d=r*1.190	16,16,11,10,8	0.90
4	86%*n/k--76%*n/k	d=r*1.185	17,16,14,13,11	0.86
5	88%*n/k--78%*n/k	d=r*1.180	17,16,14,13,11	0.86
6	90%*n/k--80%*n/k	d=r*1.175	18,17,14,14,9	0.82
7	92%*n/k--82%*n/k	d=r*1.170	18,17,14,14,9	0.82
8	94%*n/k--84%*n/k	d=r*1.165	18,17,14,13,11	0.80
9	96%*n/k--86%*n/k	d=r*1.160	19,19,17,15,9	0.79
10	98%*n/k--88%*n/k	d=r*1.155	19,19,15,14,10	0.82

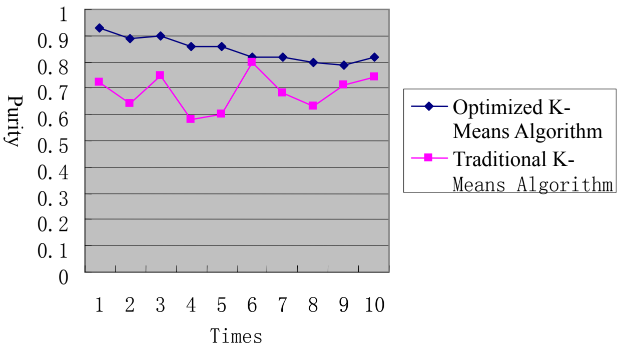

Fig.2 Comparison of Clustering Algorithm results in Purity

It can be seen from Fig.2 that although for the identical test collection, traditional k- means algorithm select clustering centers randomly. The undulation of clustering purity varies greatly, and the overall performance is unsatisfactory. The optimized algorithm has greatly improved clustering effects. As for the fixed r and d value, clustering results basically have no undulation. Even if different but appropriate r and d are selected, clustering undulation is relatively steady.

5. Conclusion

This paper has made improvement of the k-means algorithm, and presents an efficient text clustering algorithm with its aim to optimize clustering initial centers. With the purity criterion, clustering algorithm has achieved good performance in the test collection. The reason for its excellent performance is that an analysis process to each spot (namely each text object) density is performed before the k-means algorithm is conducted. This process first scans the text collection. When obtaining density size of each spot, select the best density radius and appropriate clustering central points through adjusting the step. The process provides good commencement for clustering, consequently enables the algorithm to have the possibility of jumping out partial extreme points. At the same time, the process sorted the order of text collection according to each spot density. The spot with bigger density clustered first. In so doing, the problem of sensitivity of the k-means algorithm to text input order is overcome.

References

[1] Heng Zhao, WangHai Yang. Fuzzy K-Modes Clustering Algorithm Based On Attribute Weighting[J]. Systemic Engineering and Electronic Technique, 2003.25(10):1329-1302.

[2] Yu Wang,Li Yang. An Optimized Fuzzy K-Prototypes Clustering Algorithm[J].Journal of Dalian University of Technology,2003,43(6):849-852.

[3] Tao Chen,Yan Song,YangQun Xie.Text Clustering Reserch Based On IIG And LSI Combination Characteristic Abstract[J]. Journal of the China Society for Scientific and Technical Information. 2005, 24(2): 203-209.

[4] Joshua Zhexue Huang,Michael K. Ng, Hongqiang Rong,Zichen Li .Automated Variable Weighting in k-Means Type Clustering[J], IEEE Transactions on Pattern Analysis and Maching Intelligence, 2005, 27(5):657-668.

[5] ZhiHua Wan,WeiMin OuYang,PingYong Zhang. A Dynamic Clustering Algorithm Based On Division [J]. Computer Engineering and Design,2005,26(1):177-180.

[6] STEINBACH M, KARYPIS G, KUMAR V. A Comparison of Document Clustering Techniques [R]. Department of Comp. Sci. & Eng University of Minnesota, 2000. 1-20.