Evaluation of Time Complexity Based on Max Average Distance for K-Means Clustering

ShinWon Lee[1] and WonHee Lee[2*]

[1]*Department of Computer System Engineering, Jungwon University, Chungbuk, Republic of Korea*
[2]*Department of Information Technology, Chonbuk University, Jeonbuk, Republic of Korea*
swlee@jwu.ac.kr, wony@jbnu.ac.kr

Abstract

Clustering method is used in diverse scopes, namely, information retrieve, communication security system, data mining, etc. It is divided into hierarchical clustering, partitioning clustering, and more. K-means is one of partitioning clustering. We improve performance of K-means to select initial centers of cluster through calculating rather than random selecting. This method maximizes the distance among initial centers of clusters. After that, the centers are distributed evenly and that result is more accurate than initial cluster centers selected at random. It is time-consuming, but can reduce total clustering time by minimizing the number of allocation and recalculation. We can reduce the time spent on total clustering.

Keywords: *clustering, Time complexity, K-means*

1. Introduction

Clustering method which is gathering several cluster according to special value on a large data is divided into hierarchical clustering [1][5], partitioning clustering[4][6], graph theory clustering. Mass information of modern society is limited to process data using hierarchical clustering or graph theory clustering and is inefficient to time complexity.

In this paper, we deal with K-means algorithm that is one of the partitioning clustering for mass data. It is easy to implement, if the time complexity is O(n) and the number of pattern is n. But it is too dependent on initial centers of clusters. That is, the result of clustering is different to the initial selected centers of cluster. Generally, when K-Means algorithm processes allocation and recalculation repeatedly, centers move into proper location. But if initial centers of cluster is selected and concentrated in partial area that result is not proper or the time of allocation and recalculation is increased. So we improve the performance of K-Means to select initial centers of cluster with calculating rather than random selecting. This method maximizes the distance among initial centers of cluster. After that, the centers are distributed evenly and that result is more accurate than initial cluster centers selected at random. It is time-consuming, but reduces total clustering time by minimizing count of allocation and recalculation.

In this paper, chapter 2 describes K-Means algorithm and initial center refining method of previous study. Chapter 3 proposes the method using max average distance for initial center setting method. Chapter 4 evaluates time complexity on proposed clustering method. In chapter 5, we conclude.

* Corresponding author

2. K-Means Algorithm

K-Means algorithm is the most commonly used partitioning clustering. The concept of this algorithm is to minimize the average Euclidean distance between the pattern and the pattern with the center of the cluster [3][4]. The center of cluster is the mean of the pattern belonging to the cluster or called center , and defined as follows.

$$\vec{\mu}(\omega) = \frac{1}{|\omega|}\sum_{\vec{x}\in\omega}\vec{x} \tag{1}$$

In this expression, ω is a set of patterns belonging to the cluster, $\vec{x}$ is a particular pattern belonging to the cluster. The pattern is represented as a vector with real values. The cluster is considered a sphere with the center of gravity.

RSS (Residual Sum of Squares) is a measure of how well center expresses patterns belonging to cluster, and represents the sum of squared distance of each pattern center for all patterns belonging to each cluster, and is shown in the following equation 2.

$$RSS_k = \sum_{\vec{x}\in\omega_k} \left|\vec{x} - \vec{\mu}(\omega_k)\right|^2$$

$$RSS = \sum_{k=1}^{K} RSS_k \tag{2}$$

RSS is the objective function of K-Means, this should be minimized. Figure 1 is K-Means algorithm.

$K-Means\left(\{\vec{x_1},...,\vec{x_N}\},K\right)$
1. $\left(\vec{s_1},\vec{s_2},...\vec{s_K}\right)$ // *Select Random Seeds* $\left(\{\vec{x_1},...,\vec{x_N}\},K\right)$
2. *for* $k \leftarrow 1$ *to* K
3. $\quad do\,\vec{\mu_k} \leftarrow \vec{s_k}$
4. *while stopping criterion has not been met*
5. *do for* $n \leftarrow 1$ *to* N
6. $\quad do\ j \leftarrow \arg\min_j \left|\vec{\mu_j} - \vec{x_n}\right|$
7. $\quad \omega_j \leftarrow \omega_j \cup \{\vec{x}_n\}$ // vector reallocation
8. *for* $k \leftarrow 1$ *to* K
9. $do\,\vec{\mu_k} \leftarrow \frac{1}{|\omega_k|}\sum_{\vec{x}\in\omega_k}\vec{x}$ // center recalcuration
10. *return* $\{\vec{\mu_1},...,\vec{\mu_K}\}$

Figure 1. K-Means Algorithm

It is terminated if the following conditions are available.

(1) It is repeated a predefined number of times. This condition limits the running time of clustering algorithm. But the number of iterations is not enough the quality of clustering can be reduced.

(2) The cluster belonging to the vector is repeated until it doesn't change. This condition is very good for quality of gathering except for when the cluster is small it is time consuming to focus on small clusters.

(3) It is repeated until the center is no longer changed.

(4) RSS is repeated until it drops below the threshold. When it is completed up to the standard, the quality of gathering is very good.

Actually we use the end condition that combines the method of limiting repeat numbers and repeating until threshold drops below.

3. Cluster Center Setting using Max Average Distance

In this paper, we improve the K-Means algorithm using new method on initial centers of cluster. This method should be the selected initial centers of cluster as far as possible. By doing so, the initial centers of cluster randomly selected will be biased in some areas, and this phenomenon can be prevented. And the clustering was to improve speed and the accuracy of clustering. In the proposed K-Means algorithm, a set C of the initial centers of cluster is the following equation (3).

$$C = \max \sum_{i=1}^{K} \left\| c_{avg} - c_i \right\|^2 \tag{3}$$

c_i is ith center of cluster, c_{avg} is average from c_1 to c_k .

1. Select Random *K* centers
2. *for* $x \in X$
2.1 Select *Candidate Cluster* with the closest *x*
candidate Cluster$\leftarrow$*min* $dist_{i=0,\ldots,k}(x,c_i)$
2.2 After replacing previous center by selected *candidate Cluster*, calculate new average distance

$$newDistAvg \leftarrow avg \sum_{i=1}^{k} \left| c_{avg} - c_i \right|^2$$

If c_i=*candidate Cluster* then $\left| c_{avg} - x \right|$
2.3 if *newDistAvg* > *oldDistAvg* then $c_i \leftarrow x$
3. *return* $\{c_1, \ldots, c_k\}$

Figure 2. Initial Center Setting Algorithm

Figure 3 describes setting of initial centers of cluster using two-dimensional data, when K is 3. There are c_1, c_2, c_3 centers, and new data x will look for the closest center. Comparing the distance between each center c_1, c_2, c_3 and x, we can confirm that the result is c_1. Now, put

x instead of c_1, calculate the distance $\{d'_1, d'_2, d'_3\}$ between each centers and average as follows.

$$newDistAvg = \frac{1}{K}\sum_{i=1}^{k} d'_j \tag{4}$$

$$oldDistAvg = \frac{1}{K}\sum_{i=1}^{k} d'_j \tag{5}$$

This distance can be compared with distance between the existing centers. Comparing the two average distance, newDistAvg value, substituted x for c_1 is larger value, so x is replaced by the new c_1. x ,c_2 ,c_3 are new oldDistAvg, and are a comparison of x. This process is repeated for the set X with x.

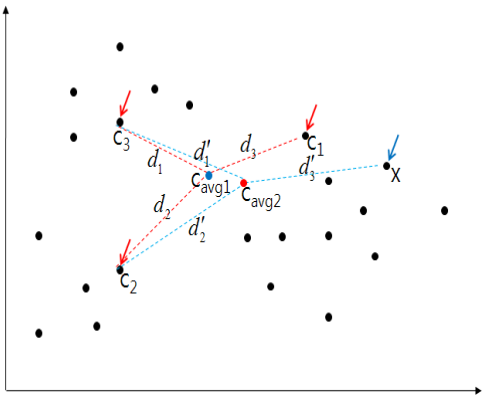

Figure 3. Initial Center Shifting using Max Average Distance

4. Evaluation of Time Complexity

Compared to existing methods of selecting centers, the method proposed in this paper requires the process of calculating the max average distance. The time required for clustering are as follows:

$$T(\text{initial center setting}) + T(\text{allocation-recalculation}) \tag{6}$$

This process takes time in addition.

In the algorithm shown in Figure 2, Step 2.1, it is 1K time to select candidate Cluster with the closest x, Step 2.2, it is 2K time to replace previous centers by x and calculate average value of centers, and 2K time to calculate the distance between average value and each centers. So the total amount of time is 5K. K is number of cluster. When time complexity of allocation- recalculation on previous K-Means algorithm is O(KN), time complexity of max average distance is as follows:

$$\div O(5KN) \tag{7}$$

The process of allocation and recalculation needs 1 unit time for allocating each documents in cluster, 1 unit time for recalculating center with documents included each cluster. The formula is as follows:

$$O(2iKN) \tag{8}$$

i is the repeated number until allocation and recalculation is finished.

So, spending time of total clustering is as follows:

$$O(5KN)+O(2iKN)\approx O(N) \tag{9}$$

i and k are constant, so it is linear time N.

$$O(5KN) \ll O(2iKN) \tag{10}$$

The time required to select the initial center doesn't have a big impact on spending time of total clustering. This should be confirmed through experiments.

5. Conclusion

In this paper, we proposed method for selection of the center to improve the performance of K-Means algorithm that is one of partitioning algorithm mainly used large amounts of data. K-Means is easy to implement, and general because time complexity is linear when the number of pattern is N. However, depending on whether or how to set the initial centers of cluster, the result of cluster is dependent on the initial centers of cluster.

We are reduced the number of allocation and recalculation process to allocate documents to each cluster and to recalculate centers. The time complexity is as follows:

$$O(5KN)+O(2iKN)\approx O(N) \tag{11}$$

It is linear for number of documents, and can reduce the time spent on total clustering. In addition, clustering result is consistent.

References

[1] G. Adami, P. Avesani and Diego Sona, "Clustering documents in a web directory", Proceedings of the 5th ACM international workshop on Web information and data management, **(2003)** pp.66-73.

[2] S. P. Lloyd, "Least squares quantization in PCM", Special issue on quantization, IEEE Trans. Inform. Theory, **(1982)** 28, pp.129-137.

[3] J. McQueen, "Some methods for classification and analysis of multivariate observations", Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, **(1967)** pp.281-297.

[4] D. A. Meedeniya and A. S. Perera, "Evaluation of Partition-Based Text Clustering Techniques to Categorize Indic Language Documents", IEEE International Advance Computing Conference(IACC 2009), **(2009)** pp.1497-1500.

[5] N. Sahoo, J. Callan, R. Krishnan, G. Duncan and R. Padman, "Incremental hierarchical clustering of text documents", Proceedings of the 15th ACM international conference on Information and knowledge management, **(2006)** pp.357-366.

[6] Y. Yonghong and B. Wenyang, "Text clustering based on term weights automatic partition", Computer and Automation Engineering (ICCAE), **(2010)** The 2nd International Conference, pp.373-377.

[7] Y.-H. Cho and G.-S. Lee, "Prediction on Clusters by using Information Criterion and Multiple Seeds", The Journal of IWIT, **(2010)** Vol. 10 No. 6, pp.153-159.

[8] J.-H. Jeon and M.-S. Kim, "A Study of Criterion for Efficient Clustering Estimation of Temporal Data", The Journal of IWIT, **(2011)** Vol. 11 No. 5, pp. 139-144.