Logical Symmetry Based K-means Algorithm with Self-adaptive Distance Metric

Wu Zu-Feng, Mu Xiao-Fan*, Liu Qiao, and Qin Zhi-guang

School of Comp. Sci. and Eng.,
University of Electronic Science and Technology of China,
Chengdu 610054, China
{wuzufeng,qliu}@uestc.edu.cn,
sdmuxiaofan@gmail.com,

Abstract. In this paper, we propose a modified version of the K-means clustering algorithm with distance metric. The proposed algorithm adopts a novel weighted Euclidean distance measure based on the idea of logical symmetry of points to its candidate clusters, which challenges the common assumption that the point similarity can only be determined by their physical distance to the centroids of the clusters. This kind of logical symmetry distance can be adaptively applied to many practical data clustering scenarios such as social network analysis and computer vision, in which the logical relationship of the clustering objectives is an important consideration in the design of the clustering algorithm. Several data sets are used to illustrate its effectiveness.

Keywords: K-means, logical symmetry, self-adaptive distance metric.

1 Introduction

Clustering problem is one of the most important questions in machine learning. Clustering is the process of partitioning or grouping a given data set into groups. This is done that data samples in the same group are similar and data samples from two different groups are dissimilar. With the development of technology and cross-disciplinary cooperation, clustering is widely used in Bioinformatics, image segment, social science. And these many clustering methods can be broadly divided in hierarchical methods and partitioning methods. [1] The K-means algorithm we talked about in this paper is a partitioning method. However, from the practical perspective, the most suitable clustering method depends on the practical problems you met.

The K-means was proposed by Stuart Lloyd in 1957 [3]. There are many advantages of the K-means algorithm, such as it is easy to implement and can be easily applied to processing the large data problem. But there are still some disadvantages like it cannot handle non-spherical clusters well, needs to specify the value of K in advance, different initial centroids may result in totally different final clusters, the clusters' size tend to be similar, and so on. And many research works has been studied to improve

* Corresponding author.

H.-Y. Jeong et al. (eds.), *Advances in Computer Science and Its Applications*,
Lecture Notes in Electrical Engineering 279,
DOI: 10.1007/978-3-642-41674-3_130,

the K-means algorithm. Baberjree and Ghost came with the goal function in K-means to get a balanced clustering result [4]; Nazeer and Sebastian improved accuracy of K-means by finding more receivable initial centroids[5][6]. Pelleg and Moore proposed the x-means algorithm based on K-means. By using Bayesian Information Criterion, x-means can efficient estimation the number of clusters [7]. But no matter how many improved algorithms have been proposed, one thing is common recognized that no one variant K-means algorithm can solve all clustering problem. Each has its own special issues. However previous researches have fewer investigations on improving the accuracy of K-means when data set has adjacent and different size clusters. So, this paper we propose a modified K-means based on weighted squared distance. The algorithm we proposed can make up the drawback of tending to partition data set of adjacent, different size clusters into similar size clusters in K-means. Experiments show that when dealing with the above situation, the modified algorithm gets higher accuracy than K-means with a tolerable gain on computation time.

The remaining of this paper is organized as follows. The second section we review the K-means algorithm and show the shortcomings. The details of modified algorithm are provided in the third section. In the forth section are the experiments and results, and in the fifth section is the conclusion. Section 6 is the acknowledgment.

2 K-means

K-means algorithm [2] is the simplest method in clustering algorithm. Given a data set $\{x_1,...,x_m\}$ and the number K. K-means divides the data set into K partition. The main idea of K-means is that: minimize the sum of squared distance from each data sample to its cluster centroid. And this is also called goal function in K-means. The formula of the goal function is as follows:

$$J = \sum_{i=1}^{m} \left\| x_i - c_{u_i} \right\|^2$$

u_i represents the cluster x_i assigned to. c_{u_i} is the centroid of the cluster u_i. In order to minimize the goal function, K-means takes following steps.

Input:

1. Data set $\{x_1,...,x_m\}$ which have m data samples.
2. Number of desired clusters, k.

Output: Data samples in the data set are divided into K clusters.

Step 1: Randomly select K data samples $c_1, c_2, ..., c_k$ from data set $\{x_1,...,x_m\}$ as initial centroids.

Step2: For each data sample $x_i, i \in (1,...,m)$ in data set, assign it to the nearest centroid.

$$u_i := \arg\min_{j \in (1,...,k)} \left\| x_i - c_j \right\|^2$$

Euclidean distance is used as distance measurement.

Step3: Recalculate centroid as the center of all data samples in the same cluster

$$c_j := \frac{\sum_{i=1}^{m} 1\{u_i = j\} x_i}{\sum_{i=1}^{m} 1\{u_i = j\}}$$

$1\{u_i = j\}$ represents that: when $u_i = j$ happens, this formula results in 1; otherwise, it results in 0. The left side c_j is the new centroid for cluster j.

Step 4: Repeat step2 and step3 until each centroid in the clusters does not change or changes less than a threshold value.

From the Step2, we saw that each time the data sample is assigned to the nearest centroid. This can cause an issue. The data sample in the bound between a larger cluster and a smaller cluster may be assigned to the smaller cluster since the data sample is far away from the larger cluster's centroid. That means K-means tend to divide the data set into clusters with comparatively similar size, which is the drawback we try to improve.

The problem comes when there are two clusters which are adjacent and have different sizes. So, we add weighted information as a compensation factor to distance measurement in the goal function of K-means. By changing the goal function, we also changing the assignment of data sample in Step2. Thus, we make up the drawback which tends to partition data set into similar size clusters in K-means. We will give a detail explanation of the modified k-means algorithm in next section.

3 Modified K-means Algorithm

The modified K-means algorithm takes the hypothesis that if there are more data samples in the cluster which are similar to the assigned data sample, the probability that the data sample be assigned to the cluster should be higher. When we say similarity, we refer to the two distances from two data sample to the same centroid are similar. And in this paper, we still use the Euclidean distance as the distance measure.

Adding weighting factor to the Euclidean distance can rectify the shortage of trending to divide the data set into similar size cluster in the K-means algorithm. And the weight is the reciprocal number of data samples which are similar to the assigned data sample. So, the more similar data samples the cluster has, the less weighted distance between data sample and cluster centroid it will be. Thus the data sample is more likely to be assigned to the cluster which has more similar data samples in it.

We use two parameters λ_1 and λ_2 to prescribe a limit to the similar between data sample pair. The distance between the assigned data sample and centroid is defined as d. If the distance between the cluster centroid and the data sample in the cluster is greater than $\lambda_1 d$ and less than $\lambda_2 d$, then, the data sample is similar to the assigned data sample.

Therefore, the modified K-means algorithm modifies the goal function in K-means to the following goal function.

$$J = \sum_{i=1}^{m} \left\| x_i - c_{u_i} \right\|^2 \frac{1}{1+\sum_{j=1}^{m} 1\left\{ \lambda_1 \left\| x_i - c_{u_i} \right\| \le \left\| x_j - c_{u_i} \right\| \le \lambda_2 \left\| x_i - c_{u_i} \right\|, u_j = u_i \right\}}$$

This function minimized the weighted sum of squared distances. $u_j = u_i$ represents that data sample x_j and x_i are in the same cluster.

Correspond to the changes in goal function; the new procedure is as follows:

Input:

1. Data set $\{x_1, ..., x_m\}$ which have m data samples.

2. Number of desired clusters, k.

Output: Data samples in the data set are divided into K clusters.

Step 1: Randomly select K data samples $c_1, c_2, ..., c_k$ from data set $\{x_1, ..., x_m\}$ as initial centroids.

Step 2: For each data sample $x_i, i \in (1, ..., m)$ in the data set, assign it to the nearest centroid based on weighted squared distance.

$$u_i := \arg\min_{j \in (1,...,k)} \left(\left\| x_i - c_j \right\|^2 \frac{1}{1+\sum_{l=1}^{m} 1\left\{ \lambda_1 \left\| x_i - c_j \right\| \le \left\| x_l - c_j \right\| \le \lambda_2 \left\| x_i - c_j \right\| \right\} 1\{u_l = j\}} \right)$$

$1\{u_l = j\}$ indicates data sample x_l has to be within the cluster j. $\lambda_1 \left\| x_i - c_j \right\| \le \left\| x_l - c_j \right\| \le \lambda_2 \left\| x_i - c_j \right\|$ represents the distance between x_l and c_j should greater than the distance between x_i and c_j multiply by λ_1 and less than that multiply by λ_2.

Step 3: Recalculate centroid as the center of all data samples in the same cluster.

$$c_j := \frac{\sum_{i=1}^{m} 1\{u_i = j\} x_i}{\sum_{i=1}^{m} 1\{u_i = j\}}$$

Step 4: Repeat step2 and step3 until each centroid in the clusters does not change or changes less than a threshold value.

4 Experimental Results

In this section, firstly, we described the data sets used in this paper and given a brief introduction to the experimental setup. Secondly, we compared the validation performance of the K-means algorithm and the modified K-means algorithms on the data sets.

4.1 Data set Description and Experimental Setup

- Synthetic data sets description

The synthetic data sets were generated by the following formula:

$$X = X_0 + r \cdot \sin\theta$$

$$Y = Y_0 + r \cdot \cos\theta$$

With the fixed (X_0, Y_0) and radius R, each data sample was generated by firstly choosing a random positive number r which follows the uniform distribution on $(0, R)$ and a random θ, $\theta \in (0^0, 360^0)$. Then calculated the corresponding X and Y. We created three data sets, each of which had two adjacent clusters. The following Table 1 gives a detail description of the synthetic data sets we used.

Table 1. Summary of synthetic data sets

Cluster	(X_0, Y_0)	Radius R	Number of data sample
Data set 1 cluster No.1	(2.0,2.0)	0.5	100
Data set 1 cluster No.2	(4.0,4.0)	2.3	400
Data set 2 cluster No.1	(2.0,2.0)	1.0	200
Data set 2 cluster No.2	(4.0,4.0)	1.8	300
Data set 3 cluster No.1	(2.0,2.0)	1.4	250
Data set 3 cluster No.2	(4.0,4.0)	1.4	250

The three data sets we finally used in the experiment are show in Figures 1. (a),(b),(c) are synthetic data set1-3, respectively.

- Experimental setup

We implemented both the K-means and the modified K-means algorithms in Python. The experimental operating system was Windows 7 with Intel Core 2Due CPU 2.0GHZ and RAM 2.0GB. We also implemented the method came from K-means++ algorithm [8] to help us find more receivable initial centroids. This cut down the bad initial centroids' effect on the accuracy. Each of the experiments was executed 10 times to make a credible result. We used parameters $\lambda_1 = 0.85$ and $\lambda_2 = 1.15$ after we had compared different parameters pairs on the data sets.

4.2 Validation Measures and Results

- Validation measures

As a Clustering algorithm, the goal of K-means is to attain high intra-cluster similarity and low inter-cluster similarity. But this is an internal criterion for the quality of a clustering algorithm. For the data sets we used in this paper, we already know the predefined classes. So, we use two external criteria, purity [9] and Rand index, to evaluate the performance of the algorithm.

The Purity is a simple and straightforward criterion. When the clustering is done, each cluster is assigned to the class which has the maximal number of data samples in the cluster. Then counting the number of correctly assigned data samples in all the clusters and dividing by the total number of data set to get the purity value. Note that some clusters may assign to the same class. And some classes may not share the maximal number of data samples with any cluster. Purity is computed use following formulate:

$$purity(A,C) = \frac{1}{N}\sum_{k=1}^{K} \max_{j\in(1,J)} \left|a_k \cap c_j\right|$$

Where $A = \{a_1, a_2, ..., a_K\}$ represents the set of clusters partitioned by the clustering algorithm and $C = \{c_1, c_2, ..., c_J\}$ represents the set of pre-defined classes. $\left|a_k \cap c_j\right|$ represents the number of data samples cluster a_k and classes c_j shared. N represents the total number of data samples in the data set.

The greater the purity value, the better performance the algorithm has.

The Rand index is motivated by classification problem. For a data set which contains N data sample, there are C_n^2 data sample pairs. And the rand index calculates the faction of correctly clustered data sample pairs to all data sample pairs. It uses a contingency table as follows:

Table 2. Contingency table

	Same cluster	**Different clusters**
Same class	TP	FN
Different classes	FP	TN

Where TP is an abbreviation for true positive, which means two data samples in the same cluster actually come from the same class. FN short for false negative, means two data samples in the different clusters indeed come from the same class. FP short for false positive which means two data sample in the same cluster come from different classes. TN short for true negative which means two data sample in the different clusters come from different classes. Formulate of Rand index is as the following:

$$Randindex = \frac{TP + TN}{TP + FP + FN + TN}$$

Like purity, the higher Rand index value indicates a better algorithm performance.

- Results

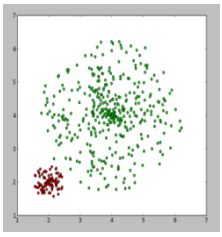
We executed both K-means and modified K-means algorithm 10 times with k=2 on three synthetic data sets. Each time the experiment was run with the same initial centroids for both algorithms. Figure 2 shows the clustering result with initial centroid ((2.415, 2.161), (5.228, 4.640)) on data set 1. On the left side (a) is the clustering

result of K-means algorithm. The data set 1 had been divided to two clusters, represented by red and green colors. We can find that the data samples on the bound of the clusters which also belong to the green cluster had been assigned to the red cluster since they are close to the red cluster centroid. This makes the original small cluster, the red cluster, larger. And the larger cluster smaller. While on the right side (b) is the clustering result of the modified K-means algorithm. The algorithm distinguished the two clusters correctly just as the original distribution on the data set.

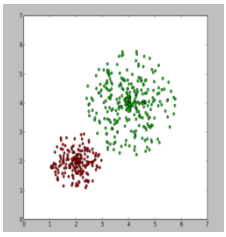
Table 3 is the average performance for both algorithms on three synthetic data sets. From performance of K-means on three data sets, we find that, the K-means get great performance on data set 3 and data set 2 but poor performance on data set 1 with average purity 0.833 and rand index 0.722. This indicated that K-means algorithm

Table 3. Purity and Rand index results on synthetic data sets

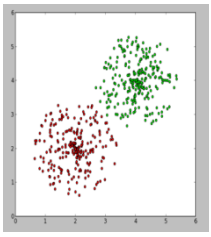
Data set	Purity		Rand index	
	K-means	***Modified K-means***	***K-means***	***Modified K-means***
Data set 1	0.833	0.958	0.722	0.932
Data set 2	0.975	0.999	0.956	0.999
Data set 3	0.999	0.997	0.999	0.995



(a) synthetic data set 1

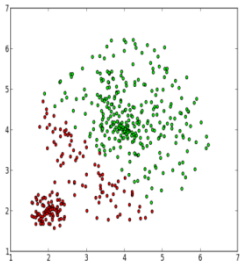


(b) synthetic data set 2

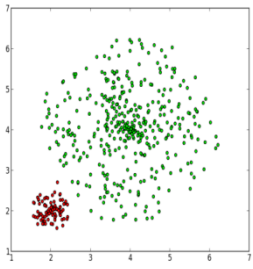


(c) synthetic data set 3

Fig. 1. Distribution of synthetic data sets. In (a), there are two clusters which are adjacent, and huge difference in size. In (b), the two adjacent clusters are fewer differences in size. And in (c), the two clusters are much similar in size.



(a) clustering result of K-means



(b) clustering result of Modified K-means

Fig. 2. Clustering result on synthetic data set1

couldn't distinguish the clusters with different size. However, the modified K-means algorithm not only performed well on data set 3 and data set 2 as K-means, but also performed great on data set 1 with average purity 0.958 and rand index 0.932. This meant the modified K-means algorithm could make up the weak performance of K-means on data set with adjacent, different size clusters.

5 Conclusion

We have proposed a modified K-means algorithm using weighted sum of squared distances as goal function. It is used to improve the clustering result of K-means when the data set has adjacent, different size clusters. In the experimental part, we generated three synthetic data sets and experimented on both algorithms. When using purity and rand index as the validation measure, the experiments result shown that both the algorithms performed well on synthetic data set which had similar size of clusters. However, when the clusters in synthetic data set had adjacent, different size clusters, the modified K-means enhanced purity with 12% and rand index with 20% than K-means algorithm did.

Acknowledgment. This work was supported by the National Science Foundation (Grant No. 61133016), the Fundamental Research Funds for the Central Universities (Grant No. ZYGX2012J067), and the National High Technology Joint Research Program of China (Grant No. 2011AA010706).

References

1. Berkhin, P.: Survey of Clustering Data Mining Techniques (2002)
2. MacQueen, J.B.: Some Methods for classification and Analysis of Multivariate Observation. In: Proceedings of 5-th Berkeley Symposium on Mathematical Statistic and Probability, vol. 1, pp. 281–297. University of California Press, Berkeley (1967)
3. Lloyd, S.P.: Least square quantization in PCM. Bell Telephone Laboratories Paper (1957)
4. Banerjee, A., Ghosh, J.: On scaling up balanced clustering algorithms. In: Proceedings of the SIAM International Conference on Data Mining (2002)
5. Abdul Nazeer, K.A., Sebastian, M.P.: Improving the Accuracy and Efficiency of the k-means Clustering Algorithm, vol. I. WCE, London (2009)
6. Chen, Z., Shixiong, X.: K-means Clustering Algorithm with improved Initial Center. In: WKDD, pp. 790–792. IEEE Computer Society, Washington, DC (2009), doi:10.1109/WKDD.2009.210
7. Pelleg, D., Moore, A.: X-means : Extending K-means with Efficient Estimation of the Number of Cluster. In: ICML 2000 Proceedings of the Seventeenth International Conference on Machine Learning, pp. 727–734. Morgan Kaufmann Publishers Inc., San Francisco (2000)
8. Arthur, D., Vassilvitskii, S.: k-means++: the advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, pp. 1027–1035 (2007)
9. Zhao, Y., Karypis, G.: Empirical and Theoretical Comparisons of Selected Criterion Functions for Document Clustering. Machine Learning 55, 311–331 (2004), doi:10.1023/B:MACH.0000027785.44527.d6