Comparative Analysis of Efficiency of k-Means and K-Medoids Algorithms for Clustering Arbitrary Data Points

[1]Neelesh Ray, [2]Sunil Phulre, [3]Vineet Richhariya

Abstract

Clustering is important data mining technique to extract useful information from various high dimensional datasets. A wide range of clustering algorithms is available in literature and still an open area for researcher. Hence in this paper we make analysis of arbitrarily distributed input data points to evaluate the clustering quality and performance of two of the partition based clustering algorithms namely k- Means and k-Medoids. To evaluate the clustering quality, the distance between two data points are taken for analysis. The computational time is calculated for each algorithm in order to measure the performance of the algorithms.

Key words: K-means Algorithm, K-medoids Algorithm, Cluster Analysis.

Introduction

Data Mining (DM) is the extraction of information from large amounts of data to view the hidden knowledge and facilitate the use of it to the real time applications. DM has a wide variety of algorithms for data analysis. Some of the major DM techniques used for analysis are Clustering, Association, Classification and etc. Clustering is an effective technique for exploratory data analysis, and has found applications in a wide variety of areas. Most existing methods of clustering can be categorized into three: partitioning, hierarchical, grid-based and model-based methods. Partition based clustering generates a partition of the data such that objects in a cluster are more similar to each other than they are to objects in other clusters. The k-Means [1, 5], EM [5], and k-medoids [6] are examples of partitional methods. Partitional algorithms have the advantage of being able to incorporate knowledge about the global shape or size of clusters by using appropriate prototypes and distance measures in the objective function [7, 8, 12, and 13]. Recently, the advent of World Wide Web search engines, the problem of organizing massive multimedia databases, and the concept of \data mining" large databases has led to renewal of interest in clustering and the development of new algorithms[9]. In many applications, clustering is used as an intermediate compression tool. First, the data is clustered, and then only the clusters' representatives are used for the analysis part. Some times the number of data objects in each and every cluster is used for analysis. But this Analysis discusses about the computational complexity of k- Means and k-Medoids algorithms. Arbitrarily distributed data points are given as input for analysis.

Methodology

Cluster analysis groups data objects based on only information found in the data that describes the objects and their relationships. The goal is that objects within a group be similar (or related) to one another and different from (or unrelated to) the objects in other groups. The greater the similarity (or homogeneity) within a group and the greater the difference between the groups are, the better or more distinct the clustering. A large number of clustering algorithms have been developed in a variety of domains for different types of applications [2, 5]. None of these algorithms is suitable for all types of applications. This research work is carried out to compare the performance of k-Means and k-Medoids clustering algorithms based on the clustering result.The basic ideas and its concepts are explored in the following sections.

The K-Means Algorithm

The k-Means is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori [10, 11]. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early group age is done. At this point it is necessary to re-calculate k new centroids as bar centers of the clusters resulting from the previous step. After obtaining these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop, one may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more. Finally, this algorithm aims at minimizing an objective function, in this case a squared error function. The objective function

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 ,$$

Where $\left\| x_i^{(j)} - c_j \right\|^2$ **is a chosen distance measure between** a data point and the cluster centre is an indicator of the distance of the *n* data points from their respective cluster centers. The algorithm is composed of the following steps:

1. Place k points into the space represented by the objects that are being clustered. These points represent initial group centroids.

2. Assign each object to the group that has the closest centroid.

3. When all objects have been assigned, recalculate the positions of the k centroids.

4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

The algorithm is significantly sensitive to the initial randomly selected cluster centers. The k- Means algorithm can be run multiple times to reduce this effect. K-Means is a simple algorithm that has been adapted to many problem domains and it is a good candidate to work for a randomly generated data points. One of the most popular heuristics for solving the k-Means problem is based on a simple iterative scheme for finding a locally minimal solution [3, 4, and 10]. This algorithm is often called the k-Means algorithm.

The K-Medoids Algorithm

The k-Means algorithm is sensitive to outliers since an object with an extremely large value may substantially distort the distribution of data [11]. Instead of taking the mean value of the objects in a cluster as a reference point, a medoid can be used, which is the most centrally located object in a cluster. Thus, the partitioning method can still be performed based on the principle of minimizing the sum of the dissimilarities between each object and its corresponding reference point. This forms the basis of the k-Medoids method. The basic strategy of k-Medoids clustering algorithms is to find k clusters in n objects by first arbitrarily finding a representative object (the medoids) for each cluster. Each remaining object is clustered with the medoid to which it is the most similar. The k-Medoids method uses representative objects as reference points instead of taking the mean value of theobjects in each cluster is the key point of this method. The algorithm takes the input parameter k, the number of clusters to be partitioned among a set of n objects. A typical k-Mediods algorithm for partitioning based on medoid or central objects is as follows:

Input: k: The number of clusters D: A data set containing n objects

Output: A set of k clusters that minimize the sum of the dissimilarities of all the objectsto their nearest medoid.

Method: Arbitrarily choose k objects in D as the initial representative objects;

Repeat assign each remaining object to the cluster withthe nearest medoid; randomly select a non

medoid object Orandom; compute the total points S of swaping object Oj withOramdom;

if S < 0 then swap Oj with Orandom to form the new set of k medoid; **Until** no change;

It attempts to determine k partitions for n objects. After an initial random selection of k medoids, the algorithm repeatedly tries to make a better choice of medoids [5, 11]. Therefore, the algorithm is often called as representative object based algorithm.

Distance Measure

An important step in most clustering is to select a distance measure, which will determine how the similarity of two elements is calculated. This will influence the shape of the clusters, as some elements may be close to one another according to one distance and farther away according to another. For example, in a 2-dimensional space, the distance between the point (x = 1, y = 0) and the origin (x = 0, y = 0) is always 1 according to the usual norms, but the distance between the point (x = 1, y = 1) and the origin can be 2, or 1 if you take respectively the 1-norm, 2-norm or infinity-norm distance. Another important distinction is whether the clustering uses symmetric or asymmetric distances. Many of the distance functions listed above have the property that distances are symmetric (the distance from object A to B is the same as the distance from B to A). In other applications, this is not the case.

A true metric gives symmetric measures of distance. The symmetric and 2-norm distance measure is used in this research work. In the Euclidean space Rn, the distance between two points is usually given by the Euclidean distance (2-norm distance). The 2-norm distance is the Euclidean distance, a generalization of the Pythagorean Theorem to more than two coordinates. It is what would be obtained if the distance between two points were measured with a ruler: the "intuitive" idea of distance. Based on this idea of finding the distance, the clustering qualities of the proposed algorithms are analyzed.

Summary

The results of both the algorithms are analyzed based on the number of data points and the computational time of each algorithm. The behavior of the algorithm is analyzed by observations. The number of data points is clustered by the algorithm as per the distribution of arbitrary shapes of the data points.

Table 1 K-Means Results

Run/Cluster		1	2	3	4	5	TOT	DT
1	Size	43	39	49	30	39	172	0
	ICT	31	15	32	32	62	172	
2	Size	28	46	29	46	51	156	8
	ICT	31	31	10	16	60	148	
3	Size	37	44	41	36	42	165	9
	ICT	31	15	31	32	47	156	
4	Size	52	45	28	37	38	162	0
	ICT	31	64	15	16	46	162	
5	Size	33	33	38	43	53	163	10

Table 2 K-Medoids Results

Run/Cluster		1	2	3	4	5	TOT	DT
1	Size	58	69	27	25	21	218	37
	ICT	15	93	31	16	16	171	
2	Size	39	41	35	47	38	219	9
	ICT	36	16	48	15	47	210	
3	Size	46	33	43	40	38	196	11
	ICT	51	47	26	26	35	185	
4	Size	38	41	44	39	38	212	10
	ICT	31	32	45	31	63	202	
5	Size	51	38	43	30	38	221	20
	ICT	59	45	40	32	25	201	

Time complexity analysis is a part of computational complexity theory that is used to describe an algorithm‟s use of computational resources; in this case, the best case and the worst case running time expressed. From table 1, the maximum and

Minimum time taken by the k-Means algorithm is 172 and 156 respectively. Like, from table 2, 221 and 196 are the maximum and minimum time taken by the k-Medoids algorithm. The performance of the algorithms have been analyzed for several executions by considering different data points (for which the results are not shown) as input (300 data points, 400 data points etc.) and the number of clusters are 10 and 15 (for which also the results are not shown), the outcomes are found to be highly satisfactory. Figure 4 shows that the graph of the average results of the distribution of data points. The average execution time is taken from the tables 1 and 2. It is easy to identify from the figure 4 that there is a difference between the times of the algorithms. Here, it is found that the average execution time of the k-Means algorithm is very less by comparing the k-Medoids algorithm.

Conclusion

From the experimental approach, for the proposed two algorithms in this research work, the obtained results are discussed. The choice of clustering algorithm depends on both the type of data available and on the particular purpose and chosen application. Usually the time complexity varies from one processor to another processor, which depends on the speed and the type of the system. The partitioning based algorithms work well for finding spherical-shaped clusters in small to medium-sized data points. The efficiency of the algorithms for the arbtrary distributions of data points is analyzed by various executions of the programs. Finally, this research work concludes that the computational time of k- Means algorithm is less than the k-Medoids algorithm for the chosen application.Hence, the efficiency of k-Means algorithm is better than the k-Medoids algorithm.

Future Work

Algorithm's comparison shows that accuracy of these algorithms is not so good for the colon dataset. However performance of global k-means and K-medoids is comparable. Performance of these algorithms can be improved. In case of k-means intial selection of cluster centres plays a very important role. So we will work on the possibility to improve these algorithms by using some good initial selection technique and fuzzy logics to achieve better results in tumor classification.

References

[1] Berkhin P, "Survey of Clustering Data Mining Techniques", Technical Report, Accrue Software, Inc, 2002.

[2] Bradley. P.S, Fayyad. U. M and Reina. C. A, Scaling "Clustering Algorithms to Large Databases", Proc. of the 4th International Conference on Knowledge Discovery & Data Mining (KDD98), AAAI Press, Menlo Park, CA, 1998, pp. 9-15.

[3] Bhukya. D. P, Ramachandram. S, Reeta Sony A L, "Performance Evaluation of Partition Based Clustering Algorithms in Grid Environment using Design of Experiments", International Journal of Reviews in Computing, vol. 4, 2010, pp. 46-53.

[4] Benderskaya. E. N and S.V. Zhukova, "Self-organized Clustering and Classification: A Unified Approach via Distributed Chaotic Computing", International Sympo-sium on Distributed Computing and Artificial Intelligence, Advances in Intelligent and Soft Computing, 2011, Vol. 91/2011, 423-431, DOI:10.1007/978-3-642-19934-9_54.

[5] Han J. and M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, Second Edition, New Delhi, 2006. ISBN : 978-81-312-0535-8

[6] Hartigan. J.A,Clustering Algorithms, Jan-1975, Wi ley Publishers, ISBN:047135645X.

[7] Jain A.K. and R.C. Dubes,Algorithms for Clustering Data, Prentice Hall Inc., Englewood Cliffs, New Jerssy, 1988. ISBN: 0-13-022278-X

[8] Jain, A.K., M.N. Murty and P.J. Flynn, 1999,"Data Clustering: A review", ACM Computing Surveys, Vol. 31, No. 3, September 1999, DOI: 10.1.1.18.2720&rep=rep1&type=pdf

[9] Kaufman, L. and P.J. Rousseeuw,"Finding Groups in Data: an Introduction to Cluster Analysis", John Wiley and Sons, 1990.

[10] Susana. A, Leiva-Valdebenito, Francisco. J and Torres-Aviles, "A Review of the Most Common Partition Algorithms in Cluster Analysis: A Comparative Study", Colombian Journal of Statistics, ISSN: 0120-1751, Vol. 33, No. 2, Dec. 2010, pp. 321- 339.

[11] Park, H.S., J.S. Lee and C.H., "A k-Means-Like Algorithm for k-Medoids Clustering and Its Performance", Department of Industrial and Management Engineering, POSTECH, South Korea, Jun, 2009.

[12] Velmurugan. T and T. Santhanam, "Computational Complexity between K-means and K-Medoids cluster-ing algorithms for normal and uniform distributions of data points", Journal of Computer Science, ISSN:1549-3636, Vol. 6, Issue 3, 2010, pp. 363-368.

[13] Velmurugan. T and T. Santhanam, "A Comparative Analysis between K-Medoids and Fuzzy C-Means Clustering Algorithms for Statistically Distributed Data Points", Journal of Theoretical and Applied Information Technology, E-ISSN 1817-3195 / ISSN 1992-8645,Vol. 27, No. 1, 2011, pp. 19–30.

Author's details

[1,2,3]Dept. of Computer Science, LNCT, Bhopal