International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 24 (2017) pp. 15147-15151
© Research India Publications. http://www.ripublication.com

Case Study on Enhanced K-Means Algorithm for Bioinformatics Data Clustering

Jasmin T. Jose[1], Ushus Zachariah[2], Lijo V.P.[3], Lydia J Gnanasigamani[4] and Jimmy Mathew[5]

[1]Assistant Professor, [2]Assistant Professor (Senior), [3]Assistant Professor, [4]Assistant Professor (Senior),
[5]Assistant Professor (Senior),

[1,2,3,4] School of Computer Science and Engineering, Vellore Institute of Technology (VIT University), Vellore, Tamil Nadu, India.

[1]Orcid Id: 0000-0001-8196-8125, [2]0000-0002-7093-7347, [3]0000-0002-5182-323, [4]0000-0002-7235-2166, [5]0000-0003-2811-1995

Abstract

In Scientific field, the data collection resulted into large scale growth by continuous additions of data. The characteristics such as velocity, variety, volume, etc. make the collected data as Big data. Analysis of such data uses various data mining techniques. Clustering is one among them and it is an unsupervised learning technique used for statistical data in bioinformatics, social networks, etc. Among various clustering techniques the K -means clustering is a common and predominant algorithm. However, the accuracy of original K-means algorithm heavily depends on centroids at the beginning and it has high computational complexity. In this paper we present an empirical study on enhanced k-means algorithm for high accuracy clustering with the initial centroids selection in an improved manner.

Keywords: Clustering Analysis, Enhanced Clustering Algorithm, k-means Algorithm

INTRODUCTION

Developments in scientific fields lead cumulative data collection and this continues addition of data increases the needs of efficient data analysis techniques. Clustering is an important and basic task in data analysis. The aim of the clustering algorithm is to group the elements in dataset into a set of meaningful disjoint subclasses. Clustering is the process of partitioning or grouping a given set of patterns into disjoint clusters [5]. Normally the patterns belongs to the same cluster are alike and pat-terns in different clusters are different. One of the predominant clustering algorithms is k-means clustering algorithm. In this algorithm select centroids of the clusters randomly. While considering huge input data, computational cost will be very high as it calculates the distance from data points to the centers for each iteration [5].

The large amount of data wants efficient storage, processing and analysis to get timely results. The data analysis is an important task in bioinformatics and social networking to discover new knowledge. Knowledge discovery from this data collection demands more efficient methods for data analysis. The algorithms based on k-means clustering helps to satisfy the users at a satisfactory level. Even though k-means clustering algorithm provides meaningful clusters of data, the accuracy of clusters is less and efficiency of the algorithm has to be improved. By the means of improving the efficiency of clustering, many research works carry outs on the k-means algorithm. A comparative analysis on k-means carried out by Kavya and Desai [13]. The k-means algorithm is compared with fuzzy c-mean algorithm, and it shows that the k-means algorithm outperforms said algorithm for the data with noise [15]. Many applications are still using k-means algorithm for analysis [14]. Verma et al. [16] claim that the k-means algorithm produces clusters with high quality and it is faster than Density Based Clustering, DBScan clustering, Hierarchical Clustering, Optics, EM Algorithm.

In this paper, present an empirical study on efficient clustering algorithm (Sort-based clustering algorithm) introduced by Nazeer et al. [12] for bio-informatics data set. This algorithm addresses the important draw backs of the k-means algorithm such as its very large time complexity. Nazeer and Sebastian [2] proposed an efficient method to select initial centroids of clusters on k-means. By this efficient method could achieve reduced time complexity as O(n2) [2]. Analysis of this algorithm reveals that the time complexity resulting from the process of selection of initial centroids. So it is desirable to develop an efficient method to find these initial centroids to get better reduced time complexity. The study reveals that the selection of initial centroids using heuristic approaches is given better accuracy than the algorithm with random initial centroids. This paper includes performance comparison between k-means algorithm and variance of heuristic approaches.

Clustering is a method for statistical data analysis used in data mining, pattern recognition and bioinformatics. Researchers in these areas were developed different types of clustering algorithms for variety of applications and huge amount of datasets. Among these clustering techniques, researchers are preferring k-means algorithm because of its simplicity, easy implementation and efficiency [6]. K-means is one of the simplest unsupervised learning algorithm for clustering of data points [9][10]. The main task of this k-means method is determining the initial centroid for each clusters. An elaborated study of different initialization methods [11] is done by M. Emre Celebi et al. A new method [3] for determining the initial centroids is presented by Yuan F. Hence it shows improved accuracy in clustering, compared to the basic k-means algorithm, as the distribution of data and obtained centroids are consistent to each other. However, the time complexity is not considered for the improvement in this method. The bad cluster initialization will results in poor quality clusters. Tzortzis and Likas [17] proposed an algorithm called as minmax k-means which assigns some weights to the clusters based on cluster variance to overcome the shortcoming of poor cluster initialization. Algorithm 1 [1] shows traditional k-means algorithm. An enhanced algorithm [2] reduces the time complexity as O(n2), and its two phases, phase I and phase II are Algorithm 2 and Algorithm 3 respectively. The phase I for finding initial centroids and phase II is assigning data points to proper clusters, are presented in [2].

In this paper, Section 2 describes the efficient K-means clustering algorithms, including sort based algorithm for initial centroid detection. Section 3 gives the Experimental results followed by the discussion on different algorithms and its efficiency in Section 4. This study work is concluded in Section 5, which also gives some future works.

Algorithm 1: k-means clustering algorithm

Input:

```
S = s1, s2,......,sn   // n data elements
k                      // k clusters
```

Output:

k cluster’s set.

Steps:

1. Select k data items out of S for initial centroid
2. Cluster each si item to the centroid with smallest average squared Euclidian distance
3. For each cluster, compute new mean value; Repeat the steps 2 and 3; Iteration ceases when a complete iteration does not move any of the cluster centers.

Different Approaches to K-means clustering

In this section we are presenting various approaches to improve the efficiency and accuracy of K-means algorithm. Algorithm 1 is modified with the complexity of O(n2), by enhancing the iterative steps of the same. In this modification, initial centroids are determined by Algorithm 2 and the clustering of data items is performed using Algorithm 3.

Algorithm 2: Initial centroid generation

Input:

```
S = s1,s2,….sn    // n data item’s set
k                 // k number of clusters
```

Output:

k initial centroid’s set.

Steps:

1. Initialize x = 1;
2. Calculate the distance among data points in S.
3. Add the pair with shortest distance to a new set Wx and remove the same from S;
4. Add new data point to the set Wx, where the new data point has shortest distance from set Wx and remove the data item from S;
5. Repeat step 4 if the number of data points in Wx less than threshold T;
6. x= x+ 1, if x less than k and Go to step 3;
7. The arithmetic mean of the sets Wx will be the initial centroids.

Calculating the distance from ith data item to all other n-1data items to identify the initial centroids is the major time consuming process in the Algorithm 2. The time complexity of this step in the above mentioned Algorithm is O(n2). After identifying the initial centroids, the cluster forming process in Algorithm 3 can be executed with a linear complexity, O(nk). A particular data item will move to other cluster, if the relative distance from that data item to its new centroid is lesser than to the old centroid, with the time complexity of O(k). Otherwise it will remain in the same cluster, with the complexity O(l). It is evident that the number of moving data points decreases with each iteration. The complexity will be O(nk/2), with assumption of 50% of data points move from clusters. Hence, the total cost of this phase of the algorithm can be expressed as O(nk) instead of O(nkl). Since the number of clusters (k) is very lesser compared to n, the total complexity in time becomes O(n2) for the enhanced algorithm.

Algorithm 3: Cluster forming

Input:

```
S =s1, s2,.....,sn        // n data items
X =x1,x2,.......,xk // k centroids
```

Output:

k clusters

Steps:

1. Compute the distance between si and xj.
2. Assign all data points to the cluster where it has shortest distance.
3. Recalculate the centroids and compute the distance of the data points from its centroid of the nearest cluster.
4. Data points stay back in same cluster if new distance is lesser or equal to old distance.
5. Else calculate its distance between new centroids and the data point.
6. Repeat from step 3 if a data point moved from one cluster to another.

Algorithm 4 gives an efficient method to determine beginning center of the clusters. In this method, initially, find the Euclidean distance of each data points from the Origin 0. The Euclidean distance is used for determining the closeness of each data point to the cluster centroids. Smaller distance indicates the belongingness of a data point to a cluster.

The distance between one vector X = (x1, x2, ...xn) and origin is obtained as,

d (X) = sqrt (x12 + x22 + … + xn2)

Then sort the distances (with data-points) in Ascending (or Descending) order by using heap sort [6]. The heap sort is used in this algorithm as it has worst, best and average case time complexity of O(n log n). Sorting is the major time consuming component in the initial centroid generation process. So the complexity of selecting initial centroid is O(n log n). Moreover, heap sort can be used to sort the large data sets. This sorted array of data-points will be divided into k parts. Then arithmetic mean of these k parts are calculated and these means are considered as the initial centroid.

The input to the Algorithm 3 is initial centroids and it assigns data-points to proper clusters with the time complexity of O(nk).

Algorithm 4: Finding the initial centroids

Input:

S=s1,s2,….sn // n data elements

k // no. of clusters

Output:

k initial centroids

Steps:

1. Set N = 0; ORIGIN = 0;
2. Compute the distance of each data points in S from the ORIGIN, put the distances in set N.
3. Sort the data points (using heap sort) in ascending order of the distance.
4. Divide the set S into k parts.
5. For each parts find the arithmetic mean of the vectors of data points, these means will be the initial centroids.

EXPERIMENTAL RESULTS

In order to analyze the performance of enhanced algorithm, some data set selected from UCI Repository [4]. The same sets are used for both the enhanced and the Original k-means algorithm. Figure 1 and Figure 2 illustrate the accuracy and efficiency of k-means and Enhanced algorithm, on testing IRIS and Ecoli data set respectively.

Table 1 illustrates the comparison of accuracy and efficiency of k-means and sort-based algorithm, which is applied on three different datasets [4] like ecoli, IRIS and Yeast data set. The result shows that the algorithm with heap sort outperforms the other two sorting algorithms, Merge and Quick Sort, where SGIs are System generated Initial Centroids.

The enhanced algorithm efficiently gives clusters with better accuracy. The tested cases give better result as depicts in Table 1.

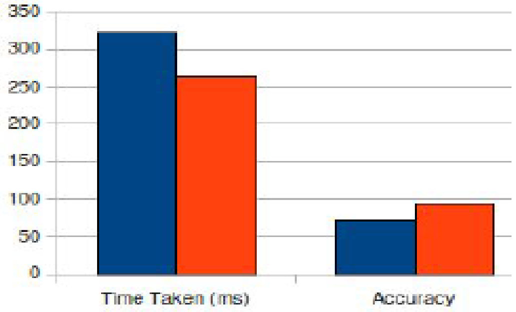

Figure 1: Efficiency and Accuracy of algorithms (on IRIS Dataset)

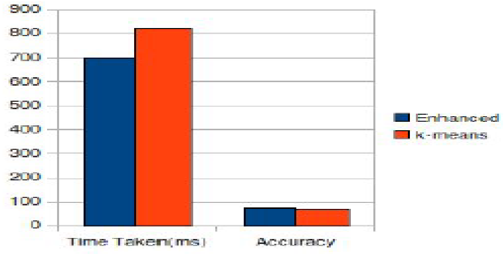

Figure 2: Accuracy and Efficiency of Algorithms (E-Coli Dataset)

Table 1: Accuracy and execution time of k-means and sort based k-means algorithm

Datasets	K-Means			Sort-based Algorithm				
	Initial Centers	Accuracy	Time Taken (ms)	Initial Centers	Accuracy	Time Taken (ms)		
						Heap Sort	Merge Sort	Quick Sort

IRIS	1.4, 4.7, 3.2, 7;
	1.4, 4.4, 3.1, 6.7;
	0.2, 1.4, 3.5, 5.1;
		87.40%	128	SGIs	89.11%	88	101	97

	0.2, 1.4, 3.5, 5.1;
	1.4, 4.7, 3.2, 7.3;
	2.5, 6.0, 3.3, 6.3;
		62.56%	132					

Ecoli	0.21, 0.67, 0.41, 0.50, 0.48, 0.49, 0.10;
	0.34, 0.61, 0.42, 0.50, 0.48, 0.51, 0.30;
	0.79, 0.77, 0.65, 0.50, 0.48, 0.42, 0.72;
	0.83, 0.81, 0.66, 0.50, 0.48, 0.41, 0.79;
	0.79, 0.76, 0.65, 0.50, 0.48, 0.48, 0.83;
	0.77, 0.74, 0.59, 0.50, 0.48, 0.43, 0.69;
	0.70, 0.82, 0.46, 0.50, 0.48, 0.36, 0.79;
		72.6%	287	SGIs	74.64%	163	199	203

	0.35, 0.56, 0.50, 0.48, 0.29, 0.49 0.29;
	0.44, 0.35, 0.54, 0.50, 0.48, 0.40, 0.07;
	0.46, 0.37, 0.49, 0.50, 0.48, 0.56 0.40;
	0.36, 0.45, 0.52, 0.50, 0.48, 0.59 0.49;
	0.35, 0.25, 0.55, 0.50, 0.23, 0.32, 0.48;
	0.46, 0.38, 0.67, 0.39, 0.48, 0.50, 0.36;
	0.34, 0.23, 0.44, 0.48, 0.29, 0.28, 0.48;
		53.09%	276					

Yeast	0.50, 0.40, 0.49, 0.39, 0.50, 0.11, 0.50, 0.00;
	051, 0.26, 0.58, 0.47, 0.54, 0.11, 0.50, 0.00;
	0.22, 0.49, 0.50, 0.34, 0.55, 0.21, 0.50, 0.00;
	0.25, 0.50, 0.61, 0.60, 0.55, 0.21, 0.50, 0.00;
	0.22, 0.50, 0.45, 0.40, 0.50, 0.16, 0.50, 0.00;
	0.22, 0.51, 0.43, 0.44, 0.48, 0.22, 0.50, 0.00;
	0.22, 0.49, 0.73, 0.63, 0.42, 0.30, 0.50 0.00;
	0.22, 0.55, 0.43, 0.53, 0.52, 0.13, 0.50, 0.00;
	0.22, 0.58, 0.46, 0.53, 0.52, 0.15, 0.50, 0.00;
	0.22, 0.54, 0.51, 0.51, 0.52, 0.51, 0.50, 0.00;
		50.3%	776	SGIs	62.17%	671	720	689

	0.22, 0.48, 0.58, 0.61, 0.47, 0.13, 0.50, 0.00;
	0.22, 0.53, 0.43, 0.47, 0.48, 0.27, 0.50, 0.00;
	0.22, 0.53, 0.64, 0.62, 0.49, 0.15, 0.50, 0.00;
	0.25, 0.54, 0.58, 0.44, 0.57, 0.13, 0.50, 0.00;
	0.22, 0.48, 0.42, 0.44, 0.48, 0.54, 0.50, 0.00;
	0.22, 0.49, 0.51, 0.40, 0.56, 0.17, 0.50, 0.50;
	0.22, 0.53, 0.50, 0.54, 0.48, 0.65, 0.50, 0.00;
	0.28, 0.58, 0.48, 0.45, 0.59, 0.20, 0.50, 0.00;
	0.22, 0.49, 0.55, 0.50, 0.66, 0.36, 0.50, 0.00;
	0.30, 0.58, 0.40, 0.39, 0.60, 0.15, 0.50, 0.00;
		67.3%	723					

Average Value:		69.048%	387		75.31%	307.33	340	329.67

DISCUSSION

In Algorithm 4, most of the time taken for sorting of the data points. But this process is done by using heap sort, which has complexity as O(n log n). So the time complexity of this algorithm is O(n log n), where n is number of data-points. Assigning each data point to the appropriate clusters, as same as Algorithm 3, with time complexity O(nk) where n is number of data-points and k is number of clusters. k is very much less than n. So the overall time complexity of the Enhanced algorithm is O(n log n). A logically improved heap sorting algorithm [7] may help to reduce the complexity further.

The Enhanced algorithm has been implemented by using Java under the Linux platform. The performance criteria of this algorithm as follows:

Accuracy: The accuracy is measured as the ratio of the number of correctly assigned data-points and the total number of data-points in the data set.

Efficiency: The computational time required by the algorithm.

CONCLUSION

The k-means algorithm is a well-known partitional clustering algorithm. But the traditional approach selects the number of clusters (k), prior to the clustering process, and randomly selected initial centroids to produce the clusters. Since the initial centroids which is selected randomly will contribute much on the accuracy of this clusters, accuracy cannot be guaranteed.

This paper aimed to present an empirical study on enhanced algorithm to produce more accurate set of clusters by computing initial centroids from the data set under consideration, rather than randomly selecting them. This computation is proved to give more accurate set of clusters without sacrificing the accuracy of the original algorithm. As the complexity of the enhanced algorithm is mostly depends on the sorting process, a logically improved heap sorting algorithm [7] can reduce the complexity further.

REFERENCES

[1] Margaret H. Dunham, Data Mining- Introductory and Advanced Concepts, Pearson Education, 2006.

[2] K. A. Abdul Nazeer, M. P. Sebastian, Improving the Accuracy and Efficiency of the k-means Clustering Algorithm, Proceedings of the World Congress on Engineering 2009 Vol I WCE 2009, July-2009, London, U.K

[3] Yuan F, Meng Z. H, Zhang H. X and Dong C. R, A New Algorithm to Get the Initial Centroids,Proc. of the 3rd International Conference on Machine Learning and Cybernetics, pages 2629, August 2004.

[4] Merz C and Murphy P, UCI Repository of Machine Learning Databases, Available: ftp://ftp.ics.uci.edu/pub/machine-learning-databases

[5] Fahim A.M, Salem A. M, Torkey A and Ramadan M. A, An Efficient enhanced k-means clustering algorithm,Journal of Zhejiang University, 10(7):1626 1633, 2006.

[6] Aggarwal CC, Reddy CK, editors. Data clustering: algorithms and applications. CRC Press; 2013 Aug 21.

[7] Cormen TH. Introduction to algorithms. MIT press; 2009 July 31.

[8] Lohani, Bhanu Prakash, et al. "Performance Study of Logically-Modified Heap Sort Algorithm." (2014).

[9] A.K. Jain, M.N. Murty, P.J. Flynn , Data clustering: A review, ACM Computing Surveys, 31 (3) (1999), pp. 264–323

[10] Data clustering: 50 years beyond k-means , Pattern Recognition Letters, 31 (8) (2010), pp. 651–666

[11] Celebi, M. Emre, Hassan A. Kingravi, and Patricio A. Vela. "A comparative study of efficient initialization methods for the k-means clustering algorithm." Expert Systems with Applications 40.1 (2013): 200-210.

[12] Nazeer KA, Kumar SM, Sebastian MP. Enhancing the k-means clustering algorithm by using a O (n logn) heuristic method for finding better initial centroids. In Emerging Applications of Information Technology (EAIT), 2011 Second International Conference on 2011 Feb 19 (pp. 261-264). IEEE.

[13] Kavya DS, Desai CD. Comparative Analysis of K means Clustering Sequentially and Parallely, International Research Journal of Engineering and Technology. 2016 Apr; 3(04).

[14] El Aziz MA, Selim IM, Essam A. Open cluster membership probability based on K-means clustering algorithm. Experimental Astronomy. 2016 May: pp 49–59.

[15] Ghosh S, Dubey SK. Comparative analysis of k-means and fuzzy c-means algorithms. (IJACSA) International Journal of Advanced Computer Science and Applications. 2013 Apr; 4(4).

[16] Verma M, Srivastava M, Chack N, Diswar AK, Gupta N. A comparative study of various clustering algorithms in data mining. International Journal of Engineering Research and Applications (IJERA). 2012 May; 2(3): 1379-84.

[17] Tzortzis G, Likas A. The MinMax k-Means clustering algorithm. Pattern Recognition. 2014 Jul 31; 47(7): 2505-2516.