Volume 3, Issue 2, February 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Performance Comparison of Various Clustering Algorithm

S. Revathi
PG Student, Department of CSE
Bharath University
Chennai, India

Dr.T.Nalini
Professor, Department of CSE
Bharath University
Chennai, India

Abstract – Clustering is the process of grouping of data, where the grouping is established by finding similarities between data based on their characteristics. Such groups are termed as Clusters. A comparative study of clustering algorithms across two different data items is performed here. The performance of the various clustering algorithms is compared based on the time taken to form the estimated clusters. The experimental results of various clustering algorithms to form clusters are depicted as a graph. Thus it can be concluded as the time taken to form the clusters increases as the number of cluster increases. The farthest first clustering algorithm takes very few seconds to cluster the data items whereas the simple KMeans takes the longest time to perform clustering.

Keywords-Clustering, Clustering algorithms, KMeans, Efficient KMeans, Filtered cluster, Make density based cluster, Farthest first.

I. INTRODUCTION

Clustering is mainly needed to organise the results provided by a search engine. Clustering can also be viewed as a special type of classification. The clusters formed as a result of clustering can be defined as a set of like elements. But the elements from different clusters are not alike. Clustering is similar to database segmentation, where like tuples in a database are grouped together. When clustering is applied to a real world database, many problems occur there such as: handling outlier is difficult; interpreting the semantics of each cluster is difficult, no correct answer for a clustering problem and what data should be used for clustering.

The problem of clustering can also be defined as below: Given a collection of data objects, the work of clustering is to divide the data objects into groups such that objects in the same group are similar. Objects in different groups should be dissimilar. Data belonging to one cluster are the most similar; and data belonging to different clusters are the most dissimilar. Clustering algorithms can be viewed as hierarchical and partitional. With hierarchical clustering, a nested set of clusters is created. The hierarchy is divided into various levels. In the lowest level, each item will have its own cluster. In the highest level, all the items will be belonging to a single cluster. With partitional clustering, only one set of cluster is created. Hierarchical clustering is represented using a tree structure called dendrogram. Examples of hierarchical clustering algorithms are agglomerative and divisive clustering algorithms. Examples of partitional clustering algorithms are KMeans, nearest neighbour and PAM. Clustering can be done on large databases also. Most popular clustering algorithms like BIRCH (balanced iterative reducing and clustering using hierarchies) [15], DBSCAN (density based spatial clustering of applications with noise) and CURE (Clustering using representatives) [16]. Clustering can also be performed with categorical attributes. Optimization based partitioning algorithms are represented by its prototype.

Objects of similar prototype are clustered together. An iterative control strategy is used to optimize the clustering. If the clusters are of convex shape, same size, density, and if their number *k* can be reasonably estimated, then the clustering algorithm can be selected correctly. *K*-means, *k*-modes and *k*-medoid algorithms can be differentiated based on their prototypes. The *k*-means method has been shown to be effective in producing good clustering results for many practical applications. However, the *k*-means algorithm requires time proportional to the product of number of patterns and number of clusters per iteration. This computationally may be expensive especially for large datasets. Among clustering formulations that are based on minimizing a formal objective function, perhaps the most widely used and studied is *k*-means clustering. Given a set of *n* data points in real *d*-dimensional space, R^d, and an integer *k*, the problem is to determine a set of *k* points in R^d, called centers, so as to minimize the mean squared distance from each data point to its nearest centre. A comparative study between various clustering algorithms based on the time taken to form the clusters is considered. The various clustering algorithms taken into consideration are simple KMeans, overlapped KMeans, enhanced KMeans are compared against filtered clusterer, make density based clusterer and farthest first clustering.

In section II, the discussion of related work is carried out. In section III, analysis of various clustering algorithms is performed. In section IV, the experiment and results of all the five clustering algorithms, with two different datasets are discussed. In section V, performances of various clustering algorithms are concluded based on the time to form the clusters, followed by the references used.

II. RELATED WORK

Based on the definition of nearest neighbour pair C. S. Li et al. [1] proposed a new cluster center initialization method for K-Means algorithm. In iterative clustering algorithms, selection of initial cluster centers is extremely important as it has a direct impact on the formation of final clusters. A simple and efficient implementation of K-Means clustering algorithm called the filtering algorithm [2] shows that the algorithm runs faster as the separation between clusters increases. Several proximity measures, cluster validation and various tightly related topics were discussed. A new generalized version of the conventional K-Means clustering algorithm which performs correct clustering without pre-assigning the exact cluster number can be found in [3]. A non-metric distance measure for similarity estimation based on the characteristic of differences [4] is presented and implemented on K-Means clustering algorithm. The performance of this kind of distance and the Euclidean and Manhattan distances were then compared.

An algorithm to compute the initial cluster centers for K-Means algorithm was given by M. Erisoglu et al. [5] and their newly proposed method has good performance to obtain the initial cluster centers converges to better clustering results and almost all clusters have some data in it. An Efficient KMeans Clustering Algorithm for Reducing Time Complexity using Uniform Distribution Data Points [6] was proposed. The accuracy of the algorithm was investigated during different execution of the program on the input data points. Finally, it was concluded that the elapsed time taken by proposed efficient K-Means is less than KMeans algorithm. Recently iterated local search (ILS) was proposed in [7]. This algorithm tries to find near optimal solution for criteria of the *k*-means algorithm. It applies *k*-means algorithm, tries to swap randomly chosen center with randomly chosen point, compares the current solution with the previous and keeps the best solution. This process is repeated a fixed number of times. Density-based approaches, apply a local cluster criterion, are very popular for database mining. Clusters are regions in the data space where the objects are dense, and separated by regions of low object density (noise). These regions may have an arbitrary shape. A density-based clustering method is presented in [8]. The basic idea of the algorithm DBSCAN is that, for each point of a cluster, the neighbourhood of a given radius (*e*), has to contain at least a minimum number of points (*MinPts*), where *e* and *MinPts* are input parameters.

Another density-based approach is WaveCluster [9], which applies wavelet transform to the feature space. It can detect arbitrary shape clusters at different scales. In [10], the density-based algorithm DenClue is proposed. This algorithm uses a grid but is very efficient, because it only keeps information about grid cells that actually contain data points, and manages these cells in a tree-based access structure. This algorithm generalizes some other clustering approaches which, however, results in a large number of input parameters. Also the density and grid-based clustering technique CLIQUE [21] has been proposed for mining in high-dimensional data spaces. Input parameters are the size of the grid and a global density threshold for clusters. The major difference from all other clustering approaches is that, this method also detects subspaces of the highest dimensionality such that high-density clusters exist in those subspaces.

III. CLUSTERING ALGORITHMS

A. *Simple KMeans Clustering*

KMeans is an iterative clustering algorithm [11], [19] in which items are moved among set of clusters until the desired set is reached. This can be viewed as a type of squared error algorithm. The cluster mean of $K_i=\{t_{i1}, t_{i2}, \ldots.., t_{im}\}$is defined as,

$$m_i= \frac{1}{m}\sum_{j=1}^{m} tij \qquad (1)$$

<u>*Algorithm 1: K-Means Clustering*</u>

```
Input: D={a_1, t_2, ….., t_m}   // set of elements.
        k                   // number of desired clusters.
Output: K              // set of clusters.
Procedure:
assign initial values for means a_1, a_2, ….a_k;
repeat
   assign each item a_i to the cluster which has the
   closest mean;
   calculate new mean for each cluster;
until convergence criteria is met;
```

In almost all cases, the simple KMeans clustering algorithm [17] takes more time to form clusters. So it is not suitable to be employed for large datasets.

B. *Efficient KMeans Clustering*

In each iteration, the *k*-means algorithm computes the distances between data point and all centers; this is computationally very expensive especially for huge datasets. For each data point, the distance can be kept to the nearest cluster. At the next iteration, compute the distance to the previous nearest cluster. By comparing the old distance with new distance, and if it is less than or equal, then the point will be in the same cluster. This saves the time required to compute distances to $k-1$ cluster centers.

Two functions are written to implement efficient KMeans clustering algorithm [12], [24]. The first is the simple KMeans, which calculates the nearest point of center. This is done by computing the distances to all centers. Each data point keeps its distance to the nearest center.

<u>*Algorithm 2:Efficient K-Means Clustering*</u>

Function distance()

```
// each point is assigned to the cluster nearby
For i=1 to b
For j=1 to k
Compute squared Euclidean distance $d^2$(ri,aj);
endfor
Find the closest centroid $a_j$ to $r_i$;
$a_j=a_j+r_i$; $b_j=b_j+1$;
MSE=MSE+$d^2(r_i, a_j)$;
Clusterid[i]=number of the closest centroid;
centroid;
endfor
For j=1 to k
$a_j=a_j/b_j$;
endfor
```

Function distance_new()

```
// each point is assigned to the cluster nearby
For i=1 to b
    Compute squared Euclidean distance
        $d^2$($r_i$, Clusterid[i]);
    If ($d^2$($r_i$, Clusterid[i])<=Pointdis[i])
        Point stay in its cluster;
Else
For j=1 to k
Compute squared Euclidean distance $d^2(r_i,a_j)$;
endfor
Find the closest centroid $a_j$ to $r_i$;
$a_j=a_j+r_i$; $b_j=b_j+1$;
MSE=MSE+$d^2(r_i, a_j)$;
Clustered[i]=number of the closest centroid;
Pointdis[i]=Euclidean distance to closest centroid;
endfor
For j=1 to k
$a_j=a_j/b_j$;
endfor
```

C. FarthestFirst Clustering

Farthest first is a variant of K Means. This places the cluster center at the point further from the present cluster. This point must lie within the data area. The points that are farther are clustered together first. This feature of farthest first clustering algorithm speeds up the clustering process in many situations like less reassignment and adjustment is needed.

D. MakeDensityBased Clustering

A cluster is a dense region of points that is separated by low density regions from the tightly dense regions. This clustering algorithm can be used when the clusters are irregular. The make density based clustering algorithm can also be used in noise and when outliers are encountered. The points with same density and present within the same area will be connected to form clusters.

Algorithm 3: Density based Clustering

1. Compute the ε-neighborhood for all objects in the data space.
2. Select a core object CO.
3. For all objects co Ɛ CO, add those objects y to CO which are density connected with co. Proceed until no further y are encountered.
4. Repeat steps 2 and 3 until all core objects have been processed.

IV. EXPERIMENT AND RESULTS

A. Experimental setup

The performance of various clustering algorithms are measured based on the time to form the clusters. Here, two datasets are used namely, letter image dataset and abalone dataset.

TABLE I DATASETS USED

Dataset	**No. of instances**	**No. of attributes**
Letter image	20,000	16

Abalone	4,177	9

Weka, a data mining tool is used to execute the datasets. Various clustering algorithms are used to form clusters and their performance evaluation is being analyzed. The two datasets used here are described below.

A.1. Letter Image dataset

The letter image dataset [21] contains statistical attributes of 20,000 digitized pictures of letters. This contains 16 attributes. The 20,000 samples are divided into 16,000 training set and 4,000 test set. Each data item is derived from the pixels of the digital letter.

A.2. Abalone dataset

The abalone's [22] age can be found by cutting the shell and counting their rings is a very tough task. The total number of instances is 4177. The number of attributes is 8. Some of the attributes used here are, sex, length, diameter, height, whole weight, etc…

A.3 Weka

Weka is a collection of open source ML algorithms used for pre-processing, classifiers, clustering, and association rule [14]. Weka is created by researchers at the University of Waikato in New Zealand. It is a Java based tool used in the field of data mining. It uses flat text files to describe the data. It can work with a wide variety of data files including its own ".arff" format and C4.5 file formats.

B. Results for datasets on different Clustering algorithms

The letter image dataset and abalone dataset are processed on various clustering algorithms such as simple KMeans, efficient KMeans, filtered clusterer, make density based clusterer and farthest first clustering.

From table II, it is shown that with a slow range of increase in the number of clusters, the time to form the clusters also increases. In the case of letter image dataset, the farthest first clustering algorithm has the shortest time to form the clusters and the simple KMeans clustering algorithm takes the longest time. In the case of abalone dataset, the farthest first clustering algorithm takes the shortest time to form clusters and the make density based clustering algorithm takes the longest time to form the clusters.

TABLE II TIME TAKEN TO FORM THE RESPECTIVE NUMBER OF CLUSTERS

Data set	**No. of clusters**	**Time (in secs)**				
		Simple kmeans	**Enhanced kmeans**	**Farthest first**	**Make density based**	**filtered**
Letter image	**40**	34.4	13	2.79	40.25	31.03
	50	47.64	22	2.1	52.69	52.17
	60	39.02	38	4.55	36.83	36.62
	70	33.16	65	2.87	37.61	36.48
	80	48.25	48	4.72	70.36	22.53
	90	57.66	68	5.04	46.18	67.99
	100	79.4	68	4.96	61.08	52.69
Abalone	**100**	6.18	2	0.72	7.32	6.82
	200	6.86	4	1.42	11.19	11.84
	300	16.75	7	2.26	15.43	14.45
	400	15.68	8	3.09	14.9	11.56
	600	21.09	9	4.49	17.83	19.61
	800	18.05	10	5.38	22.17	19.55
	1000	19.48	12	6.47	23.35	20.61

C. Graph representation for performance evaluation

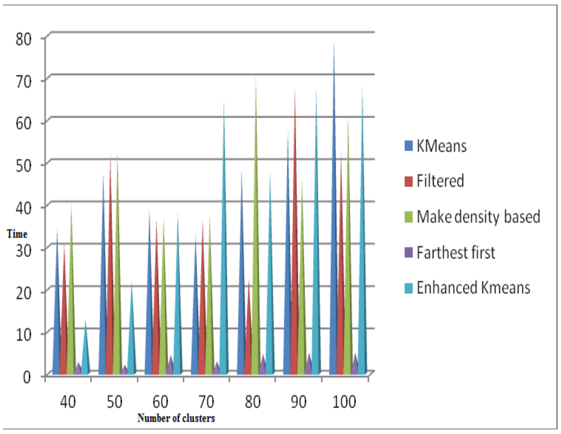

Figure 1 Execution time (letter image dataset)

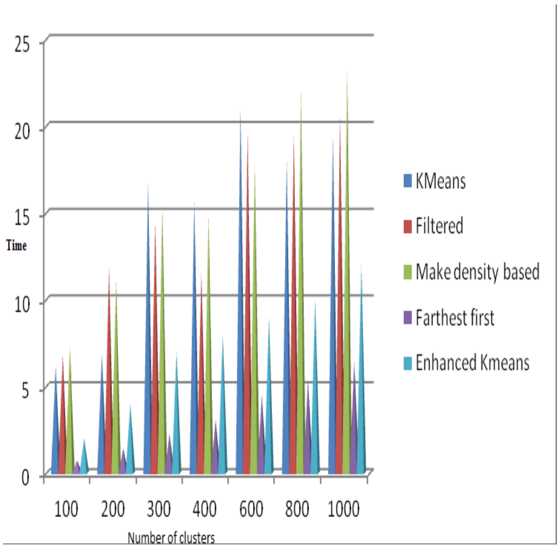

Figure 2 Execution time (abalone dataset)

V. CONCLUSION

A comparative study of clustering algorithms across two different data items is performed here. The performance of the various clustering algorithms is compared based on the time taken to form the estimated clusters. The experimental results of various clustering algorithms to form clusters are depicted as a graph. As the number of cluster increases gradually, the time to form the clusters also increases. The farthest first clustering algorithm takes very few seconds to cluster the data items whereas the simple KMeans takes the longest time to perform clustering. Thus it is very difficult to use simple KMeans clustering algorithm for very large datasets. This proposal can be used in future for similar type of research work.

ACKNOWLEDGEMENT

The authors would thank the management of Bharath University for their support and the department of Computer science and engineering for their encouragement towards the research work.

REFERENCES

[1] C. S. Li, “Cluster Center Initialization Method for K-means Algorithm Over Data Sets with Two Clusters”, “2011 International Conference on Advances in Engineering, Elsevier”, pp. 324-328, vol.24, 2011.

[2] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko and A. Wu, “An Efficient K-Means Clustering Algorithm: Analysis and Implementation”, “IEEE Transactions on Pattern analysis and Machine intelligence”, vol. 24, no.7, 2002.

[3] Y.M. Cheung, “A New Generalized K-Means Clustering Algorithm”,“Pattern Recognition Letters, Elsevier”,vol.24,issue15, 2883–2893, Nov.2003.

[4] Z. Li, J. Yuan, H. Yang and Ke Zhang, “K-Mean Algorithm with a Distance Based on the Characteristic of Differences”, ”IEEE International conference on Wireless communications, Networking and mobile computing”, pp. 1-4, Oct.2008.

[5] M. Erisoglu, N. Calis and S. Sakallioglu, “A new algorithm for initial cluster centers in K-Means algorithm”, “Published in Pattern Recognition Letters”, vol. 32, issue 14, Oct.2011.

[6] D. Napoleon and P. G. Laxmi, “An Efficient K-Means Clustering Algorithm for Reducing Time Complexity using Uniform Distribution Data Points”, “IEEE Trendz in Information science and computing”, pp.42-45, Feb.2011.

[7] Merz, P., 2003. An Iterated Local Search Approach for Minimum Sum of Squares Clustering. IDA 2003, p.286-296.

[8] Ester, M., Kriegel, H.P., Sander, J., Xu, X., 1996. A Density- based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining. AAAI Press, Portland, OR, p.226-231.

[9] Sheikholeslami, G., Chatterjee, S., Zhang, A., 1998. Wave- Cluster: A Multi-Resolution Clustering Approach for Very Large Spatial Databases. Proc. 24th Int. Conf. on Very Large Data Bases. New York, p.428-439.

[10] Hinneburg, A., Keim, D., 1998. An Efficient Approach to Clustering in Large Multimedia Databases with Noise. Proc. 4th Int. Conf. on Knowledge Discovery and Data Mining. New York City, NY.

[11] Jain, A.K., Dubes, R.C., 1988. “Algorithms for Clustering Data”. Prentice-Hall Inc.

[12] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu, "An Efficient k-Means Clustering Algorithm: Analysis and Implementation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, VOL. 24, NO. 7, PP. 881-892, 2002.

[13] Eduardo Raul Hruschka, Ricardo J. G. B. Campello, Alex A. Freitas, and Andre C. Ponce Leon F. de Carvalho ,” A Survey of Evolutionary Algorithms for Clustering”, *IEEE Trans. Syst., Man, Cybern.—Part C: Appl. And Review,* Vol. 39, No. 2,PP.133-155,2009.

[14] Jiawei Han, Micheline Kamber,”Data Mining:Concepts and Techniques”,Second Edition,*Elesvier Publications*,2006.

[15] Zhang, T., Ramakrishnan, R., Linvy, M., 1996. BIRCH: “An Efficient Data Clustering Method for Very Large Databases”. Proc. ACM SIGMOD Int. Conf. on Management of Data. ACM Press, New York, p.103-114.

[16] Guha, S., Rastogi, R., Shim, K., 1998. CURE: An Efficient Clustering Algorithms for Large Databases. Proc. ACM SIGMOD Int. Conf. on Management of Data. Seattle, WA, p.73-84.

[17] Shi Na, L. Xumin, G. Yong, “Research on K-Means clustering algorithm-An Improved K-Means Clustering Algorithm”, “IEEE Third International Symposium on Intelligent Information Technology and Security Informatics”, pp.63-67, Apr.2010.

[18] R. Xu and D. Wunsch, “Survey of Clustering Algorithms”, “IEEE Transactions on Neural networks”, vol. 16, no. 3, May 2005.

[19] Joshua Zhexue Huang, Michael K. Ng, Hongqiang Rong, and Zichen Li,” Automated Variable Weighting in k-Means Type Clustering”, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* VOL. 27, NO. 5, PP. 657-668, 2005.

[20] Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P., 1998. Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. Proc. ACM SIGMOD Int. Conf. on Management of Data. Seattle, WA, p.94-105.

[21] http://archive.ics.uci.edu/ml/datasets/Letter+Recognition

[22] http://archive.ics.uci.edu/ml/datasets/Abalone

[23] Yedla M, Pathakota SR and Srinivasa TM (2010) Enhancing K-means clustering algorithm with improved initial center. Intl Journal of Computer Sci and Info Tech 1 : 121–125.

[24] Fahim AM, Salem AM, Torkey FA, Ramadan MA (2006) An efficient enhanced k-means clustering algorithm. Journal of Zhejiang University SCIENCE A7:1626–1633. Available online at www.zju.edu.cn/jzus.