ISSN: 0974-2115
www.jchps.com Journal of Chemical and Pharmaceutical Sciences

An efficient clustering algorithm for mining high speed data streams

N. Sivakumar[1*], Dr. S. Anbu[2]

[1] Department of Computer Science and Engineering, St. Peter's University, Avadi, Chennai, Tamilnadu, India.

[2] Department of Computer Science and Engineering, St. Peter's college of Engineering and Technology, Avadi, Chennai, Tamilnadu, India.

***Corresponding author: E-Mail:sivakumarn002@gmail.com**

ABSTRACT

It is an idea that aims to find the essential structure of data by grouping the data objects into similar groups. The high speed data stream is a data that arrives fast, continuously and transient. In order to combine the data structure safely the efficient clustering algorithm is required. In this paper we proposed an Efficient Clustering (EC) algorithm to analyze the streaming data. Also we made the comparative analysis with the popular algorithms to produce the essential features of an efficient clustering algorithm.

KEY WORDS: Data Mining, Clustering, Efficient Clustering Algorithm, Hierarchical clustering, Agglomerative Clustering, High speed data streams, Data stream clustering.

1. INTRODUCTION

Data stream is an arranged continues data items where the sequence of input arrives in time basis. There are many applications in which the high speed data streams are produced such as telecommunication call logs, sensory inputs, network monitoring, clustering of stock prices etc. The high speed data stream system that has been producing large amounts of data.

Recently the researchers focused on algorithms which are suitable for huge data sets. Mining high speed data stream is a way to bring up the hidden knowledge from fast arriving data streams, it is a way to acquire knowledge from streams and find the growth of a stream over time. The various number of methods like sliding window, sampling, sketching, synopsis data structures, load shedding etc., can be used to translate streaming data into a particular form for data analysis. The streaming data analysis techniques such as classification, regression, clustering and so on are used.

In data streams the huge amounts of data are arriving rapidly, these are very difficult to process and storage aspects and these are raises a new challenges in the research problems. Surely they are not possible to store the continuous arriving data in the traditional storage management systems. Preferably the stream data have been processed from online so that the results must be up-to-date and it must be processed using simple query method with time constraint.

High speed data: High speed data can change the behavior of data in the case of clustering. When data arrives fast, the stream speed increases from low grade to high grade so that the clustering data makes fast. Suppose when data arriving decreases from high grade to low grade then the stream data will become imperfect. Therefore the problem in clustering must be analyzed.

The further study of this paper is as follows: Section 2 contains the background study of this paper both on streaming data and clustering. Section 3 contains proposed methodology, Section 4 contains various clustering methods, Section 5 contains about EC algorithm, Section 6 contains the comparisons, Section 7 contains experimental results and Section 8 contains the conclusion of the paper.

Related work: The different methods for finding clusters of random shape have been proposed. The DBSCAN method could handle only the static environment but not able to handle the fast changing streams. Data stream models are not possible for random access. The data streams models are varying from the relational model (Charu, 2009).

Various reasons of variations of data streams models: The incoming data are not under the system control. Mining data stream systems are latent of unbounded size. Stream data is very difficult to retrieve unless it is stored in memory. Stream files may be archived or discarded. Stream has limited memory, time constraint and produces approximate results by processing them. The stream data must be scan at once and clustering via local information (Babcock, 2002; Chandrika1, 2012).

There are different methods for finding insight clusters of random shape have been proposed in the literature. These methods assume that all the data are occupant on hard disk and one can get general information about the data at any time. Thus they are not applicable for processing data streams (Sun Jigui, 2008).

The recent work constitutes ranking based method that evolves the k-means clustering algorithm execution and exactness. In that the k-means clustering algorithm have been analyzed by 2 ways, one is have an existence k-means approach that was contained with some starting point(threshold) values and second one is ranking method applied on k-means algorithm and also equate the functioning of these two methods by using graphs (Navjot Kaur, 2012)

Proposed methodology:

High speed data: This can change the behavior of data in the case of clustering. When data arrives fast, the stream speed increases from low grade to high grade so that the clustering data makes fast.
Find the Similarity between objects: This step helps you to calculate the distance between the objects using some functions. These functions also support to compute the measurement.
Group the objects using EC algorithm: This step helps you to link pairs of the objects that are in close proximity using the linkage functions. The linkage function uses the distance information which have generated in the above to find out the proximity of objects to each other. As objects are coupled binary clusters, the newly shaped clusters are sorted into larger clusters until a hierarchical tree is formed.
Output Cluster: This is the last step that helps you to use the cluster function to eliminate branches away the bottom of the hierarchical tree and allocate the objects below for each cut to a single cluster. This creates a partition of the data. The cluster function makes groupings in the hierarchical tree by separating off the hierarchical tree at an absolute point.
Various Clustering methods: Clustering is a method that groups the object of the datasets of the same group and it is one of the methods in mining data stream. The task of mining data streams are complex for certain types of clustering. The clustering algorithm is used to arrange the data into many similar to the group or dissimilar to other group. In order to mining the data streams the efficient algorithm is required.

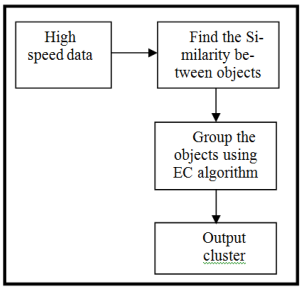

Figure.1.Proposed methodology diagram

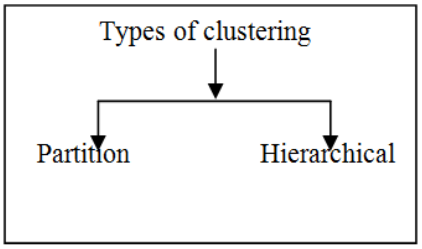

Figure.2.Cluster types

Partition clustering - *K*-means: *K*-means algorithm constructs *k* partitions of n data. The larger cluster appears to be separated into a lower difference area and a higher difference area. This may point that the larger cluster is two intersection clusters. We use k-means technique to compute the distances from each one centroid to point on a grid.
Distance squared Euclidean: It is the method of every centroid is the mean of the points in that cluster. The Formula is $d(x,c) = (x-c)(x-c)'$ (1)
where d is the distance; x and c are the points of the cluster.
Sum of differences: Each one centroid is the element wise median of the points in that cluster. The formula is

$$d(x,c) = \sum_{j=1}^{p} [x_j c_j] \quad (2)$$

where d is the distance; j is the indicator function; x and c are different point in the cluster.
Deciding of n clusters: It is an idea of how well-classify the output clusters are could be known by silhouette plot method. The silhouette plot displays an amount of how close for each one point in the nearby clusters. This ensures that the ranges from +1, identifying points that are very distance from nearby clusters, through 0, suggested point are probably allotted to the wrong cluster.
K-means local minima: In numerical minimizations, often the k-means technique reaches to starting point. It is possible for k-means to attain local minima, where allocating any one point to a new cluster that would high the total sum of point to centroid but where a best solution exists. However using 'duplicates' one can defeat the problem by taking the one with the lowest total amount of distances, over all duplicates as the final answer.
The Process of k-means:
Figure 3 explains that the process of k-means and it is used for large datasets. It is non-supervised learning algorithm, simple and applied to solve the problem related to long familiar clusters. It is the type of partitioning clustering, classify the given data objects into k clusters by iterative method so that it becomes independent. Centroid involves finding the average vector location for every clusters and bringing the distance between centroids and calculate the distances. .
Algorithm for *K* -means clustering method
Step1: Input
K number of clusters
Datasets D={d_1,d_2,d_n} //[d-distance]
Step2: Output
Set of k clusters
Step3: Repeat
Form k clusters and assign the centroid

Calculate the distance between each object
Re-compute the centroid of each cluster
Until the centroid do not change

In the above algorithm, the database is assigned into k clusters in which each register to be owned by the nearest average value cluster. This starts with some collection of data then the mean value for each cluster will calculate.

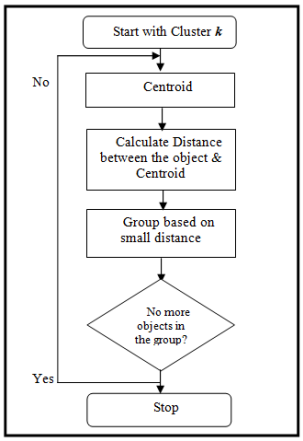

Figure.3.Flow chart for the process of K-means

EC Algorithm (Efficient Clustering): EC is an algorithm that is richer to identify the clusters having with non domain data. EC is an Agglomerative hierarchical clustering algorithm. It is an idea to ensure nearby points end up in the same cluster and tested using different threshold values. This starts with a collection C of n singleton clusters; each cluster contains one data point: $C_i\{X_i\}$

Efficient Clustering (EC) Algorithm

Step1: Input

$X=\{x_1,x_2,\ldots\ldots x_n\}$ //$[x_1 \ldots x_n]$ object sets

Step2: Output:

dis_func(c_1,c_2) // Distance function

Step3: Loop: i=1 to n

$c_i=\{x_i\}$

End Loop

Step4: $C=\{c_1,c_2,\ldots c_n\}$ // Clusters

l=n+1 // length l

Step5: Loop until C.size>1 do

(C_{min1},C_{min2})=min_dist(C_i,C_j) //for all C_i,C_j in c

delete C_{min1},C_{min2} from C

insert {C_{min1},C_{min2}) to C

l=l+1

End loop

Proximity matrix: This determines the distance between each cluster using a distance function and update every time. The following methods are used to measure the distances between each clusters.

A. Single Linkage
B. Average linkage
C. Complete Linkage

Single Linkage: In this method the distance between 2 clusters can be defined as the shortest distance between 2 points in each cluster. For example in figure 3 the distance between clusters r and s to the left is equal to the length of the arrow between their 2 closest points

Formula: $SL(r,s)=\min(D(x_n,x_{ij}))$ (3)

Where SL is Single Linkage, r and s are the clusters, D is represent minimum distance and x_n,x_{ij} are various objects.

Average Linkage: In this method the distance between 2 clusters can be defined as the intermediate distance of each point in one cluster to every point in the other cluster. For example in figure 4 the distance between clusters r and s to the left is equal to the intermediate length of each arrow between connecting points of one cluster to the other.

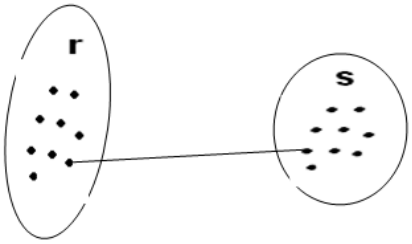

Figure.4. Single Linkage

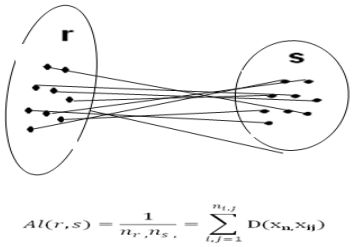

Figure.5. Average Linkage

Formula:

$Al(r,s) = \frac{1}{n_{r,}n_{s,}} = \sum_{i,j=1}^{n_{i,j}} D[x_n x_{ij}]$ (4)

Where AL is Average Linkage, r and s are the clusters, D is the distance and x_n,x_{ij} are various objects.

Complete Linkage: In this method the distance between 2 clusters can be defined as the longest distance between 2 points in each cluster. For example in figure 5 the distance between clusters r and s to the left is equal to the length of the arrow between their 2 furthest points.

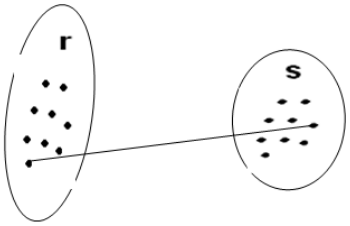

Figure.6. Complete Linkage

Formula: $CL(r,s)=\max(D(x_n,x_{ij}))$ (5)

Where CL is Complete Linkage, r and s are the clusters, D is represents Maximum distance and x_n,x_{ij} are various objects.

Example:

Euclidean function = $\sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2}$ (6) Where X and Y are vector directions

Comparison between *K*-means and EC algorithm

Scan at once: This is an important constraint because of data arrives fast and huge. There is no to read the same data again for computation in the *k*-means but EC algorithm performs this fast.

Memory & CPU usage: in *K*-means, the system can be able to process very large data in CPU but it has limited memory size. In EC algorithm these files can be archived.

Streaming data vs. cluster: The user has no previous knowledge about the data clustered and its condition. In addition, with streaming data the cluster alters the process over time.

Ability of an algorithm:

- Find an absolute attributes of clusters without an existing knowledge,
- Filters noise in streaming data so that clustering result can be good.
- This algorithm maintains fixed number of clusters over a modern applicable part of the stream for memory constraint.

Cluster compactness: Generally the system is being mined huge amount of data in which all the data cannot be clustered because of limited memory and limited space. Thus using EC algorithm we mined over 20000 observations with limiter size of memory.

Experimental result: To study the execution of the proposed EC algorithm, we applied and tested on MATLAB 2015A with 2 GB of main memory. We used Iris database for cluster the efficiency of an algorithm and also used various diagrams as an output. According to the information was given above, the maximum number of top degree nodes keeps increasing until the total number of items reaching to one. To run this experiment the following dataset was taken in the web. www.math.uah.edu/stat/data/Fisher.txt

Dendrograms: The EC algorithm that builds a cluster hierarchy which is generally shown as tree diagram called a Dendrograms.

In the figure 7, the Dendrogram starts with each object in a separate cluster. In each step, the 2 clusters that are common are merged into a single new cluster. The horizontal axis represents the objects in the original data set. The height of head 'U' represents the distance of various objects. For example in the above figure 7, object 2 groups with object 1, 3, 4 & 5 with the height of 3.5.

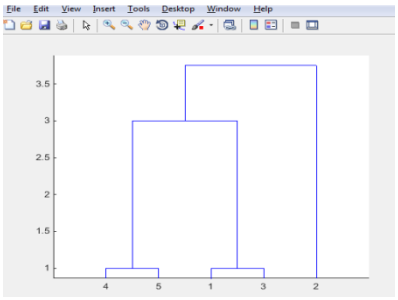

Figure.7.Screen shot of Dendrogram

In the figure 8 of binary cluster tree, column 1 and column 2 are connected in pairs to build a binary tree with first 15 values of 20000 observations. The leaf nodes are counted from 1 to n. Leaf nodes are the set clusters from which all higher clusters have built. There are m-1 higher clusters which equate to the interior nodes of the clustering tree. The third column contains the linkage distances between the two clusters merged in row.

Cluster data with more than 20000 observations.

```
firstfifteen =

  102.0000  143.0000         0
    8.0000   40.0000    0.1000
    1.0000   18.0000    0.1000
   10.0000   35.0000    0.1000
  129.0000  133.0000    0.1000
   11.0000   49.0000    0.1000
    5.0000   38.0000    0.1414
   20.0000   22.0000    0.1414
   30.0000   31.0000    0.1414
   58.0000   94.0000    0.1414
   81.0000   82.0000    0.1414
  117.0000  138.0000    0.1414
    9.0000   39.0000    0.1414
    4.0000   48.0000    0.1414
   28.0000   29.0000    0.1414

fx >>
```

Figure.8.Screen shot of Binary Cluster Tree

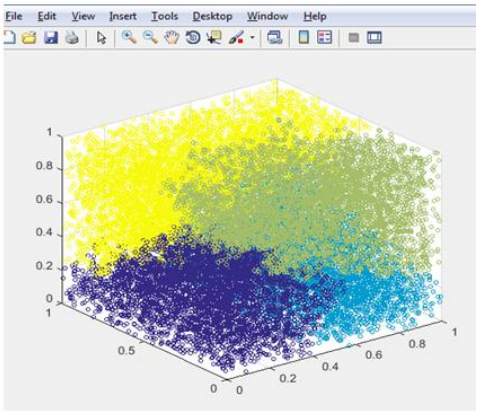

Figure.9.Screen shot of clustering the scatter diagram

In the figure 9, the randomly generated sample data with more than 20000 observations was clustered. Thus there are 4 different clustering colors can be seen. These are represents about the clustering data storage in memory. Suppose if the memory is in off mode the system will produce the status of out-of-memory.

2.CONCLUSION

Efficient Clustering (EC) is an enhancement of agglomerative hierarchical clustering algorithm and it is widely used for clustering large sets of data. From evaluation of the experiments, we can conclude that the accuracy of EC algorithm for iris datasets bearing real attributes is better than the k-means and simple hierarchical clustering methods. The time taken for clustering was very less. Generally this algorithm uses dendrogram and scatter diagrams. These are the good clustering methods that produce high quality clusters. As we have discussed in this paper this algorithm performs mining data streams is much better for large datasets.

REFERENCE

Amrita A, Kulkarni**,** Human Genome Data Clustering Using K-Means Algorithm" International Journal of Computing and Technology, 1(4), 2014.

Babcock B, Babu S, Datar M, R Motwani and Widom J Models and issues in data stream systems, Proceedings of PODS, 2002.

Chandrika1 J, Ananda Kumar K.R**,** Dynamic Clustering Of High Speed Data Streams, IJCSI International Journal of Computer Science Issues, 9(2), 2012.

Charu C, Aggarwal, A Framework for Clustering Massive-Domain Data Streams, In Proc. Of IEEE International Conference on Data Engineering DOI 10.1109/ICDE, 2009, 13.

Cormode G, Fundamentals of Analyzing and Mining Data Streams, Workshop on Data Stream Analysis, 2007, 15-16.

Deshani KAD, AN Exploratory Analysis On Half-Hourly Electricity Load Patterns Leading To Higher Performances In Neural Network Predictions, International Journal of Artificial Intelligence & Applications (IJAIA), 5(3), 2014.

Madjid Khalilian, Norwati Mustapha, Data Stream clustering: Challenges and issues, Proceedings of the International Multi Conference of Engineers and Computer Scientists, IMECS, Kong, 1, 2010, 17 – 19.

Navjot Kaur, Jaspreet Kaur Sahiwal, Navneet Kaur, Efficent K-means Clustring Algorithm Using Ranking Method In Data Mining, International Journal of Advanced Research in Computer Engineering & Technology, 1(3), 2012.

Nidhi Singh, International Journal of Computer Science and Information Technologies, 3(3), 2012, 4119-4121.

Sun Jigui, Liu Jie, Zhao Lianyu, Clustering algorithms Research, Journal of Software, 19(1), 2008, 48-61.