The Clustering in Large Databases using Clustering Huge Data Sets (CLHDS) Algorithm

Rajesh Tirlangi,Ch.V.Krishna Mohan,P.S.Latha Kalyampudi,G.Rama Krishna

[*] Department of Computer Science and Engineering, Malla Reddy College of Engineering for women, JNTUH, Hyderabad, INDIA

Abstract- Clustering is the unsupervised classification of patterns (data items) into groups (clusters).Clustering in data mining is very useful to discover distribution patterns in the underlying data. Today, the term "a large dataset" refers to hundreds of terabytes or even petabytes of data. This type of datasets are too difficult to for a clusters. It is typical of scientific investigations to have two phases: the data generation phase, and the data analysis phase. The data generation phase is usually the result of running a large simulation or the collection of data from experiments. It is desirable to design an ant colony optimization algorithm (ACO)[6][4][5] that is not required to solve any hard sub problem but can give nearly optimal solutions for data clustering. The proposed method can obtain optimal solutions quicker via differently favorable strategy. In this paper, we present a new data clustering method for data mining in large databases. Our simulation results show that the proposed Clustering huge datasets(CLHDS) method performs better than the Fast SOM combines K-means approach (FSOM+K-means) and Genetic K-Means Algorithm (GKA,K-Medoids algorithm.

Index Terms- Clusters、datamining、CLHDS、k-means

I. Introduction

Cluster analysis is the process of finding groups of object in such a way that the objects of a group are similar (or related) to one another and different from (or unrelated to) the objects of the other groups. The clustering problem[1] has been addressed in many contexts and by researchers in many disciplines; this reflects its broad appeal and usefulness as one of the steps in exploratory data analysis. Therefore given a set of data points(each with a set of attributes) with a similarity measures among them, we try to find out the groups or clusters such that 1)Data points in same cluster are similar to each other and 2) Data points in different clusters are very different from each other. Clustering approaches [1] aim at partitioning a set of data points in classes such that points that belong to the same class are more alike than points that belong to different classes. These classes are called clusters and their number may be reassigned or can be a parameter to be determined by the algorithm. Cluster analysis is used in wide variety of areas such as psychology, social sciences, biology, pattern recognition, information retrieval, machine learning and data mining etc. There exist applications of clustering in such diverse fields as business, pattern recognition, communications, biology, astrophysics and many others. Clustering methods are mainly suitable for the investigation of interrelationships between samples to make a preliminary assessment of the sample structure. Clustering techniques are required because it is very difficult for humans to intuitively understand data in a high-dimensional space. The major clustering methods can be classified into five categories: Hierarchical, Partitioning, Density-based, Grid-based, Model-based methods.

Euclidian or manhattan distance measures are most commonly used in clustering algorithms to determine clusters. Algorithms which are based on these measures generally find spherical shape clusters with similar size and density. Typical clustering algorithms work nicely on relatively smaller data sets. But huge data sets may contain millions of data objects[4]. If we cluster on a sub set of huge data sets, it may result in biased results. Most of the algorithms require user to input values for certain parameters for cluster analysis. Outliers or missing, unknown, or erroneous data are common in most real world applications. Clustering algorithms are susceptible to such data and may lead to clusters of poor quality.

$d(p,q)=\sqrt{}(p_1-p_1)^2+(p_2-q_2)^2+\ldots+(p_n-q_n)^2$ (Euclidean distance)[6]
$d(p,q)=(|\ p_1-p_1|+|p_2-q_2|+\ldots+|p_n-q_n|\ )$ (Manhattan distance)[6]

Where d(p,q) is distance between two points **p** and **q** as the length of the line segment **pq.**

Prototype-based partitional clustering algorithms can be divided into two classes: crisp clustering where each data point belongs to only one cluster, and fuzzy clustering where every data point belongs to every cluster to a certain degree . Fuzzy clustering algorithms can deal with overlapping cluster boundaries. Partitional algorithms are dynamic, and points can move from one cluster to another. They can incorporate knowledge regarding the shape or size of clusters by using appropriate prototypes and distance measures. Most partitional[6] approaches utilize the alternating optimization techniques, whose iterative nature makes them sensitive to initialization and susceptible to local minima. Two other major drawbacks of the partitional approach are the difficulty in determining the number of clusters, and the sensitivity to noise and outliers.

The k-means algorithm is a centroid based portioning technique. The k-means algorithm attempts to classify the given data sets or observations into k clusters. The k-medoids is a representative object-based technique. The partitioning method performs based on the principle of minimizing the sum of the dissimilarities between each object and its corresponding reference point.

Partitioning Around Medoids (PAM) is more robust than k-means in the presence of noise and outliers. It is efficient for small data sets but does not scale well for large data sets.

Hierarchical clustering is static, and points committed to a given cluster in the early stages cannot move to a different cluster.

Hierarchical clustering does not partition data into a particular cluster in a single step. Instead, there is a series of portioning which may run from a single cluster containing all objects to n clusters each containing a single object. Hierarchical clustering is subdivided into two categories: Agglomerative, divisive methods.

Clustering can be generally defined as the following problem. Given *N* points in *d* dimensional feature space, find interesting groups of points. Many algorithms assume that the number of clusters, *k*, is known a priori and find the *k* clusters that minimize some error metric[10]. CLHDS can work for comparatively huge databases then PAM[1]. Like PAM, CLHDS also tries to find k representative objects which are placed at the center in the cluster. This can be achieved internally by taking into account data subsets of fixed size, therefore the overall computation time and space requirements become linear in the total number of objects instead of quadratic. On the other hand, PAM consumes $O(n^2)$ memory as the collection of all a pair-wise distances between objects is required to be stored. Therefore PAM and CLARA are not suitable for larger values of n. To overcome this problem CLHDS does not compute the entire dissimilarity matrix at a time. CLHDS only computes the actual measurements(n*p data matrix).

Clustering of objects is performed in two steps in CLHDS. Initially, it picks an example from the set of objects and divides it into k clusters, using the same approach as in PAM. The two parts of algorithm are- BORN and SWAP. The CLHDS has also been applied with success to other combinatorial optimization problems such as the scheduling, partitioning, coloring, telecommunications networks, vehicle routing problem, Traveling Salesman Problem (TSP)[6].

II. Definitions for Clustering Problem

A clustering *C* means partitioning a data set into a set of clusters *Ci*, *i* = 1,…, *H*. A widely adopted definition of optimal clustering is a partitioning that minimizes distances within and maximizes distances between clusters. Within- and between-clusters distances can be defined in several ways; is widely utilized with SOM (Self-Organizing Feature Map). In addition, the *k*-means error criterion is based on it. In order to evaluate the proposed method, we define the time cost for clustering as follows,

$$T_a = \sum \frac{(T_s - T_e)}{r_n}$$

where *Ta* represents the time cost for clustering, *rn* denotes the number of runs, *Ts* is the initial time for clustering, *Te* represents the terminate time of clustering.

III. Ant Colony Optimization (ACO)

The ant colony optimization technique has emerged recently as a novel meta-heuristic belongs to the class of problem-solving strategies derived from natural (other categories include neural networks[5], simulated annealing, and evolutionary algorithms) system where low level interactions between single agents (i.e.,artificial ants) result in a complex behavior of the whole ant colony. Ant system optimization algorithms[3] have been inspired by colonies of real ants, which deposit a chemical substance (called pheromone) on the ground. It was found that the medium used to communicate information among individuals regarding paths, and used to decide where to go, consists of pheromone trails. A moving ant lays some pheromone (in varying quantities) on the ground, thus making the path by a trail of this substance. While an isolated ant moves essentially at random, an ant encountering a previously laid trail can detect it and decide with high probability to follow it, thus reinforcing the trail with its own pheromone. The collective behavior where that emerges is a form of autocatalytic behavior where the more the ants following a trail, the more attractive that trail becomes for being followed.

Given a set of n cities and a set of distances between them, the Traveling Salesman Problem (TSP) is the problem of finding a minimum length closed path (a tour), which visits every city exactly once. We call *dij* the length of the path between cities *i* and *j*. An instance of the TSP is given by a graph (*N*, *E*), where *N* is the set of cities and *E* is the set of edges between cities (a fully connected graph in the Euclidean TSP). The process is thus characterized by a positive feedback loop, where the probability with which an ant choose a path increases with the number of ants that previously chose the same path.

IV. Clustering Huge Data Sets

In this paper, CLHDS can work for comparatively huge databases then PAM. Like PAM, CLHDS also tries to find k representative objects which are placed at the center in the cluster. Therefore PAM and CLARA are not suitable for larger values of n, to overcome this problem CLHDS does not compute the entire dissimilarity matrix at a time. CLHDS only computes the actual measurements (n*p data matrix).CLHDS algorithm has two parts are – BORN and SWAP[6].

BORN: In BORN, successive medoids are selected to get the Smallest possible average distance between the objects of given sample and its most similar representative objects.

SWAP: In SWAP, an attempt is made to reduce the average distance by replacing representative objects. After that every object which is not belonging to the sample is assigned to the nearest medoid. This produces a clustering of all objects.

The quality of clustering is defined by the average distance between each object and its medoid. This procedure is repeated five times and clustering with the lowest average distance is retained for further analysis. The final average distance, the average and the maximum distance to each medoid are calculated in the same way as in PAM for all objects. Also, the ratio of the maximum distance of the medoid to the maximum distance of the medoid to another medoid. This ratio gives information on the tightness of a cluster. A small value (0.2) indicates a very tight cluster, while a value>1 indicates a weak cluster.

The CLHDS algorithm can be formally given as

```
INPUT
        D= { $T_1, T_2, ……..T_n$} //set of elements
        A      //Adjacency matrix
        K      // number of desired clusters.
OUTPUT
        K      // set of clusters
```

The k-means algorithm[6] attempts to classify the given Data sets or observations into k clusters. The k mean algorithm is iterative in nature. Let $x_{1,}x_{2......}x_n$ are data points and each data point will be assigned to one and only one cluster. It will use as Euclidian distance for dissimilarity measure. The iterative method is repeated until the function does not converge.

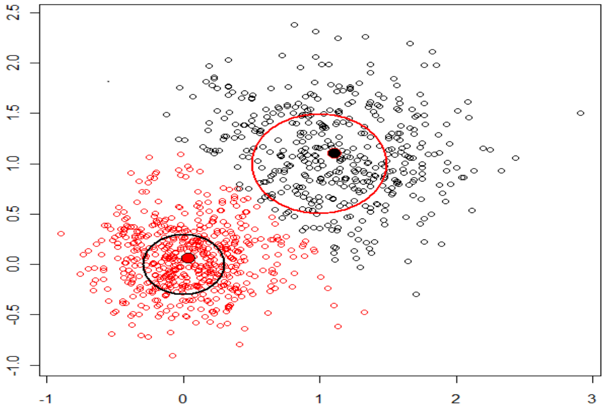

Fig : 1. Clustering using K-means

Above figure shows that k-means arbitrarily choose two objects as the two initial clusters centers, where cluster centers are marked by a " o ". Each object is distributed to a cluster based on the cluster center to which it is nearest.

Recall that our problem is to cluster data sets that are too large to fit into memory. One solution to this problem is to divide the original data set into smaller pieces which will fit into memory, cluster them, and obtain an approximation to each piece of data. Once this is done, the approximations can be gathered into one system and then clustered. A more formal description of the method follows:

A matrix M can be divided into ks disjoint sections such that:

M= [M1M2 . . . Mks],

where each section Mj is n ×kd. The partitioning of M is assumed to be virtual or arbitrary (e.g. only the data for one section is in memory at a given time), and the ordering of the columns of M is assumed to be unimportant, Once a section Mj is available, an approximation to Mj can be constructed:

Mj ≈ CjZj ,

Each column of Zj has at most kz nonzeroes. The centroids in each Cj are obtained through some kind of clustering algorithm.

Once an approximate representation is available for each section of data, they can be assembled into the approximate low memory factored representation of the entire data set M:

M ≈ CMZM,

Where

CM = [C1 C2 . . . Cks] (an n × kskc matrix)

4.1 Clustering using Object-Based Technique

In this method we pic actual objects to represent cluster instead of taking the mean value of the objects in cluster as a reference point. Then every remaining object is clustered with the representative object to which it is the most similar. The partitioning method performs based on the principle of minimizing the sum of the dissimilarities between each object and its corresponding reference point. CLHDS is handles outliers well because an object with an extremely large value may substantially distort the distribution of data.

CLARA[1] has a fixed sample at each stage of the search, CLHDS picks a sample with some randomness in each step of the search. The clustering process can be viewed as a search through a graph, where each node is a potential solution. Two nodes are neighbors if their sets differ by only one object. Each node can be assigned a cost that is defined by the total dissimilarity between every object and the medoid of its cluster. At each step, PAM examines all of the neighbors of the current node in its search for a minimum cost solution. The current node is then replaced by the neighbor with the largest descent in value. Because LARA works on a sample of nodes at the beginning of a search of the entire data sets[6], CLARANS[1] also work like as CLARA but it can search step by step entire data set. CLHDS dynamically draws a random sample of neighbors in each step of search. The number of neighbors to be randomly sampled is restricted by a user specified parameter.

CLHDS does not confine the search to a localized area. If a better neighbor is found(i.e., having a lower error), CLHDS moves to the neighbor's node and the process starts again; otherwise, the current clustering produces a local minimum. If local minimum is found, CLHDS starts with the new randomly selected nodes in search for a new local minimum. Once a user-specified number of local minimum(i.e., having the lowest cost).

The following points are weakness of k-means algorithm

- Need to specify *k*, the *number* of clusters, in advance
- Unable to handle noisy data and *outliers*
- Not suitable to discover clusters with *non-convex shapes*

These problems are overcomes the CLHDS method.

The following points are weakness of CLARA[6] method, these problems also overcome in this paper.

- Efficiency depends on the sample size.
- A good clustering based on samples will not necessarily represent a good clustering of the whole data set if the sample is biased.

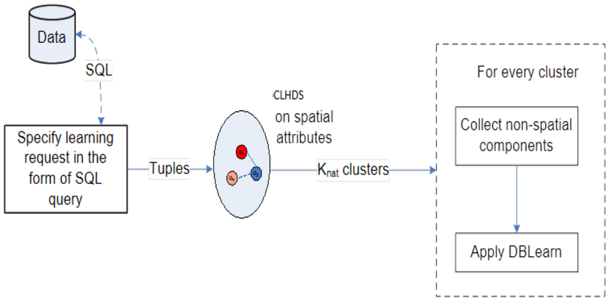

Fig: 2. Find non-spatial clusters using CLHDS

V. Conclusions

In this paper, we propose a clustering algorithm called Clustering Huge Data sets (CLHDS) with different favor for other data clustering In this paper, we present a new data clustering method for data mining in large databases. CLHDS does not confine the search to a localized area. If a better neighbor is found(i.e., having a lower error), CLHDS moves to the neighbor's node and the process starts again; otherwise, the current clustering produces a local minimum. Through experiments, we show that CLHDS efficiently finds accurate clusters in large high dimensional datasets[7] than k-means, PAM, CLARA.

References

[1] Hichem Frigui and Raghu Krishnapuram,"A robust competitive clustering algorithm with applications in computer vision," IEEE Transactions of Pattern Analysis and Machine Intelligence, vol. 21, no. 5, pp.450-465, May 1999.

[2] Sudipto Guha , Rajeev Rastogi and KyuseokShim,"CURE: An efficient clustering algorithm for largedatabase," Information Systems (Elsevier Science), vol.26, no. 1, pp. 35-58, 2001.

[3] Randall S. Sexton and Robert E. Dorsey, "Reliableclassification using neural networks: a genetic algorithm and backpropagation comparision," Decision Support System, vol. 30, pp. 11-22, 2000.

[4] Hiroshi Ishikawa , Manabu Ohta and KokiKato,"Document Warehousing: A document-intensive application of a multimedia database," Eleventh International Conference on Data Engineering, pp. 25-31, 2001.

[5] L. Lucchese and S.K. Mitra, 1999 IEEE InternationalConference on Content-Based Access of Image andVideo Libraries, pp. 74 -78, 1999.

[6] Sunita Tiwari and neha chaudary "data mining and warehousing" Dhanpat Rai&co.

Authors

First Author – Rajesh Tirlangi, Department of Computer Science and Engineering, Malla Reddy College of Engineering for Women, JNTUH, Hyderabad, India
Second Author – Ch.V.Krishna Mohan, Department of Computer Science and Engineering, Malla Reddy College of Engineering for Women, JNTUH, Hyderabad, India
Third Author – P.S.Latha Kalyampudi, Department of Computer Science and Engineering, Malla Reddy College of Engineering for Women, JNTUH, Hyderabad, India
Fourth Author – G.Rama Krishna, Department of Computer Science and Engineering, Malla Reddy College of Engineering for Women, JNTUH, Hyderabad, India