ABSTRACT

Data mining can be comprehended as a process of extraction of knowledge hidden in extremely large datasets. Data mining techniques can significantly boost the ability to analyze the data.

Clustering is one of the important techniques in data mining. Clustering becomes an indispensable requirement while dealing with immense volume of data. Clustering algorithms partition a dataset into several groups such that the intra cluster similarity is maximized and the inter cluster similarity is minimized. Existing clustering algorithms, such as K-means, PAM, CLARANS, DBSCAN, CURE, and ROCK are designed to find clusters that fit some static models. These algorithms can breakdown if the choice of parameters in the static model is incorrect with respect to the data set being clustered, or if the model is not adequate to capture the characteristics of clusters.

K-means Clustering is an important algorithm for identifying the structure in data. K-means is the simplest clustering algorithm. This algorithm is based on distance minimization between assumed cluster center and data points assigned to the cluster. This algorithm takes a predefined number of clusters and an acceptable threshold of errors between ideal convergences to actual convergence as input. Then a set of records equal to predefined numbers of clusters are chosen randomly and an iterative process towards converging the cluster centers will be carried out till an acceptable error rate defined by the threshold is achieved. Though simple in nature, K-means algorithm has some limitations.

1) When the clustering is used as an unsupervised learning algorithm for data mining, it is difficult to provide the number of clusters apriori as the structure is not known.
2) Since the records are chosen randomly the order of the records determine the quality of the clusters, which requires repetitive runs to get a better set of cluster centers. This increases the search space and time.

3) Since the records are chosen randomly the time for convergence for a solution with acceptable error rate is non-deterministic and can be quite high.

In addition most of the implementations of the standard K-means algorithm are based on data structures which are assumed to fit into memory. Working on larger datasets requires sampling or data reduction in some form which may lead to information loss if not properly tuned.

In order to minimize some of the limitations of the K-means algorithm, in this research a novel approach for finding the cluster centers has been proposed. Subsequently the K-means algorithm has been implemented in a declarative language provided by the DBMS to provide an ability to handle massive datasets that cannot fit into memory.

The novel seeding algorithm is based on an assumption of potential possibility of existence of gaps in an ordered sequence of an attribute value set. Once these gaps are identified and statistically validated for sufficient differentiation they will be projected across other attributes which may not have such kind of properties. The empirical experimentation on synthetic and real world datasets indicate this approach can substantially improve the quality of clusters by means of both identifying the structure of the clusters in the data and providing initial cluster centers which are near optimal. Since the proposed algorithm works on the basis of sorting the data it can be easily implemented in a declarative language like SQL.

SQL implementation of K-means algorithm has been tested on row stored database and found to perform exceptionally well on large datasets. However no work has been done in evaluating performance of K-means algorithm with column stored
database on large datasets. In this research an empirical evolution of the proposed extended K-means algorithm has been carried out. The results indicate superiority of column store architectures in some areas of computation while row oriented architectures score in some area.

However one of the limitations of the current research is that the implementation has been tested only on numerical datasets which lend themselves well for ordered sequencing or sorting. Use of non-numeric attribute is handled through representation of categorical attributes as numeric attributes through flattening.