ISSN 2047-3338

Partition Algorithms– A Study and Emergence of Mining Projected Clusters in High-Dimensional Dataset

K. Naveen Kumar[1], G. Naveen Kumar[2] and Ch. Veera Reddy[3]
[1]St. Mary's Engineering College (JNTUH), India
[2,3]SCIENT Institute of Technology, India

***Abstract*– High-dimensional data has a major challenge due to the inherent sparsity of the points. Existing clustering algorithms are inefficient to the required similarity measure is computed between data points in the full-dimensional space. In this work, a number of projected clustering algorithms have been analyzed. However, most of them encounter difficulties when clusters hide in subspaces with very low dimensionality. These challenges motivate effort to propose a reliable K-mediods partitional distance-based projected clustering algorithm. The proposed process is based on the K-Means and K-Mediods algorithm, with the computation of distance restricted to subsets of attributes where object values are dense. K-mediod algorithm is capable of detecting projected clusters of low dimensionality embedded in a high-dimensional space and avoids the computation of the distance in the full-dimensional space. Our research article is based on analysis of the effective performance of K-mediods.**

***Index Terms*– Data Mining, Clustering, K-Means, K-Mediods and Outlier**

I. INTRODUCTION

DATA mining is a convenient way of extracting patterns, which represents mining implicitly stored in large data sets and focuses on issues relating to their feasibility, usefulness, effectiveness and scalability. It can be viewed as an essential step in the process of knowledge data discovery. Data are normally preprocessed through data cleaning, data integration, data selection, and data transformation and prepared for the mining task. Data mining can be performed on various types of databases and information repositories, but the kind of patterns to be found are specified by various data mining functionalities like class regression, association, classification, prediction, cluster analysis etc.

Although data mining is a technology, companies are using powerful computers to shift through volumes of supermarket scanner data and analyze market research reports for years. However, continuous innovations in computer processing power, disk storage and statistical software are dramatically increasing the accuracy of analysis, while driving down the cost we need to depend on data mining tools.

Clustering data arises in many disciplines and has a wide range of applications. Intuitively, the clustering problem can be described as follows: Let W be a set of n data points in a multi-dimensional space. Find a partition of W into classes such that the points within each class are similar to each other. To measure similarity between points a distance function is used. A wide variety of functions has been used to define the distance function. The clustering problem has been studied extensively in machine learning [1], [2] databases [3], [4] from various perspectives and with various approaches and focuses.

Most clustering algorithms do not work efficiently in high dimensional spaces due to the curse of dimensionality. It has been shown that in a high dimensional space, the distance between every pair of points is almost the same for a wide variety of data distributions and distance functions [7]. Feature selection techniques are commonly utilized as a preprocessing stage for clustering, in order to overcome the curse of dimensionality. The most informative dimensions are selected by eliminating irrelevant and redundant ones. Such techniques speed up clustering algorithms and improve their performance [6].

Nevertheless, in some applications, different clusters may exist in different subspaces spanned by different dimensions. In such cases, dimension reduction using a conventional feature selection technique may lead to substantial information loss [7].

A. Plan of the Paper

This paper focuses on a detailed introduction about mining and clustering in high dimensional data. Section 2 describes Clustering techniques. Section 3 describes the proposed system and comparative study. Section 4 describes usage of K-Mediods algorithms. Section 5 finally concludes the articles.

II. CLUSTERING

Cluster is a number of similar objects grouped together. It can also be defined as the organization of dataset into homogeneous and/or well separated groups with respect to distance or equivalently similarity measure. Cluster is an aggregation of points in test space such that the distance between any two points in cluster is less than the distance between any two points in the cluster and any point not in it. There are two types of attributes associated with clustering, numerical and categorical attributes. Numerical attributes are associated with ordered values such as height of a person and speed of a train. Categorical attributes are those with unordered values such as kind of a drink and brand of car.

Clustering is available in flavors of

- Hierarchical
- Partition (non Hierarchical)

In hierarchical clustering the data are not partitioned into a particular cluster in a single step. Instead, a series of partitions takes place, which may run from a single cluster containing all objects to n clusters each containing a single object [12].

Hierarchical Clustering is subdivided into agglomerative methods, which proceed by series of fusions of the n objects into groups, and divisive methods, which separate n objects successively into finer groupings.

A. K-Means Clustering

Unsupervised K-means learning algorithms that solve the well known clustering problem. The procedure follows to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other.

The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k new centroids as bary centers of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move anymore. Finally, this algorithm aims at minimizing an *objective function*, in this case a squared error function. The objective function is as follows:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^j - c_j \right\|^2$$

Where $\left\| x_i^j - c_j \right\|^2$ is a chosen distance measure between a data point x_i^j and the cluster centre c_j is an indicator of the distance of the n data points from their respective cluster centers.

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the K centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

B. K-Medoids Algorithm

The K-means algorithm is sensitive to detect outliers since an object with an extremely large value may substantially distort the distribution of data. How might the algorithm be modified to diminish such sensitivity? Instead of taking the mean value of the objects in a cluster as a reference point, a Medoid can be used, which is the most centrally located object in a cluster. Thus the partitioning method can still be performed based on the principle of minimizing the sum of the dissimilarities between each object and its corresponding reference point. This forms the basis of the K-Medoids method.

The basic strategy of K-Mediods clustering algorithms is to find k clusters in n objects by first arbitrarily finding a representative object (the Medoids) for each cluster. Each remaining object is clustered with the Medoid to which it is the most similar. K-Medoids method uses representative objects as reference points instead of taking the mean value of the objects in each cluster. The algorithm takes the input parameter k, the number of clusters to be partitioned among a set of n objects.

A typical K-Mediods algorithm for partitioning based on Medoid or central objects is as follows:

Input:

K: The number of clusters

D: A data set containing n objects

Output: A set of k clusters that minimizes the sum of the dissimilarities of all the objects to their nearest medoid.

Method: Arbitrarily choose k objects in D as the initial representative objects,

Repeat: Assign each remaining object to the cluster with the nearest medoid;

Randomly select a non medoid object Orandom; compute the total points S of swap point Oj with Orandom

if S < 0 then swap Oj with Orandom to form the new set of k medoid

Until no change

III. PROBLEM DEFINITION

However, with the high-dimensional data commonly encountered nowadays, the concept of similarity between objects in the full-dimensional space is often invalid and generally not helpful. Recent theoretical results that data points in a set tend to be more equally spaced as the dimension of the space increases, as long as the components of the data point are independently and identically distributed.

A number of projected clustering algorithms have been proposed. However, most of them encounter difficulties when clusters hide in subspaces with very low dimensionality. These challenges motivate our effort to

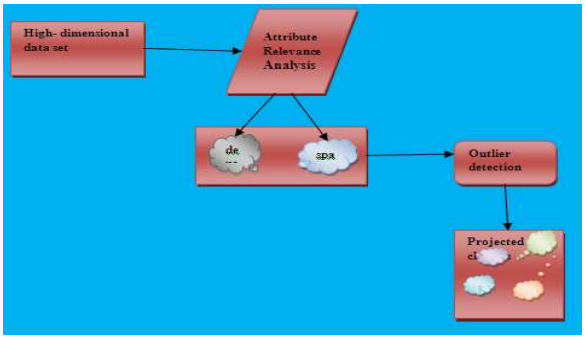

Fig. 1. This idea of the system is taken from (Ref. [11])

propose a robust partitional distance-based projected clustering algorithm. The algorithm consists of three phases. The first phase performs attribute relevance analysis by detecting dense and sparse regions and their location in each attribute. Starting from the results of the first phase, the goal of the second phase is to eliminate outliers, while the third phase aims to discover clusters in different subspaces.

A. Comparative Study

Partition algorithms K-means and K-medoid work well for finding spherical-shaped and subspace clusters in small to large data points. K-means algorithm is its favorable execution time and the user has to know in advance how many clusters are to be searched, k-means is data driven is efficient for smaller data sets and anomaly detection. Instead of taking the mean value of the objects in a cluster as a reference point, a Medoid can be used, which is the most centrally located object in a cluster.

Thus the partitioning method can still be performed based on the principle of minimizing the sum of the dissimilarities between each object and its corresponding reference point. This forms the basis of the K-Medoids method, characteristic feature is that it requires the distance between every pair of objects only once and uses the distance at every stage of iteration. Compare to two partition algorithms k-medoid is better performs for large datasets, noise and outlier detection.

IV. ANALYSIS

Finding projected clusters has been addressed in [8]. The partitional algorithm PROCLUS, which is a variant of the K-medoid method, iteratively computes a good medoid for each cluster. With the set of medoids, PROCLUS finds the subspace dimensions for each cluster by examining the neighboring locality of the space near it. After the subspace has been determined, each data point is assigned to the cluster of the nearest medoid. PROCLUS and ORCLUS were the first to successfully introduce a methodology for discovering projected clusters in high-dimensional spaces, and they continue to inspire novel approaches.

A. PROCLUS

PROCLUS [8] is the top-down subspace clustering algorithm. Similar to CLARANS [9], ROCLUS samples the data, then selects a set of k-medoids and iteratively improves the clustering. The algorithm uses a three phase approach consisting of initialization, iteration, and cluster refinement. Initialization uses a greedy algorithm to select a set of potential medoids that are far apart from each other. The objective is to ensure that each cluster is represented by at least one instance in the selected set. The iteration phase selects a random set of k-medoids from this reduced dataset, replaces bad medoids with randomly chosen new medoids, and determines if clustering has improved. Cluster quality is based on the average distance between instances and the nearest medoid. For each medoid, a set of dimensions is chosen whose average distances are small compared to statistical expectation. The total number of dimensions associated to medoids must be k * l, where l is an input parameter that selects the average dimensionality of cluster subspaces. Once the subspaces have been selected for each medoid, average Manhattan segmental distance is used to assign points to medoids, forming clusters. The medoid of the cluster with the least number of points is thrown out along with any medoids associated with fewer than (N/k) * min Deviation points, where min Deviation is an input parameter.

The refinement phase computes new dimensions for each medoid based on the clusters formed and reassigns points to medoids, removing outliers. Like many top-down methods, PROCLUS is biased toward clusters that are hyper-spherical in shape. Also, while clusters may be found in different subspaces, the subspaces must be of similar sizes since the user must input the average number of dimensions for the clusters.

However, using a small number of representative points can cause PROCLUS to miss some clusters entirely. The cluster quality of PROCLUS is very sensitive to the accuracy of the input parameters, which may be difficult to determine.

B. ORCLUS

ORCLUS [6] is an extended version of the algorithm PROCLUS [5] that looks for non-axis parallel subspaces. This algorithm arose from the observation that many datasets contain inter-attribute correlations. The algorithm can be divided into three steps: assign clusters, subspace determination, and merge. During the assign phase, the algorithm iteratively assigns data points to the nearest cluster centers. The distance between two points is defined in a subspace E, where E is a set of orthonormal vectors in some d-dimensional space. Subspace determination redefines the subspace E associated with each cluster by calculating the covariance matrix for a cluster and selecting the orthonormal eigenvectors with the least spread (smallest Eigen Values). Clusters that are near each other and have similar directions of least spread are merged during the merge phase. The number of clusters and the size of the subspace dimensionality must be specified. The authors provide a general scheme for selecting a suitable value.

A statistical measure called the cluster sparsity coefficient is provided which can be inspected after clustering to evaluate the choice of subspace dimensionality. The algorithm is computationally intensive due mainly to the computation of covariance matrices. ORCLUS uses random sampling to improve speed and scalability and as a result may miss smaller

clusters. The output clusters are defined by the cluster center and an associated partition of the dataset, with possible outliers.

C. Attribute Relevance Analysis

In the mining projected cluster, irrelevant attributes contain noise/outliers and sparse data points, while relevant ones may exhibit some cluster structure [5]. By cluster structure, a mean region that has a higher density of points than its surrounding regions. Such dense region represents the 1D projection of some cluster. Hence, it is clear that by detecting dense regions in each dimension are able to discriminate between dimensions that are relevant to clusters and irrelevant ones.

D. Outlier Detection

Data objects that do not comply with the general behavior or model of the data. Such data objects, which are grossly different from or inconsistent data with the remaining set of data, are called outliers.

The outliers may be of particular interest, in the case of fraud detection, where outliers may indicate fraudulent activity. Thus, outlier detection and analysis is an interesting data mining task, referred to as outlier mining or outlier analysis.

V. CONCLUSION

The analysis of article is a robust distance-based projected clustering algorithm for the challenging problem of high dimensional dataset clustering, and illustrated the efficient performance of K-Mediods algorithm and comparisons with K-Means work. Finally this work proposes a system of mining projected clusters and survey on solutions, which are suitable algorithms for the system in the paper.

Future work of this article extends with subspace projected partitional clustering algorithms in place of PROCLUS and ORCLUS and dissimilarity measure implementation of mining projected major clusters.

REFERENCE

[1]. P. Cheeseman, J. Kelly, and M. Self. “Auto Class: A bayesian classification system”. In ICML’88, 1988.

[2]. P. Smyth, “Probabilistic model-based clustering of multivariate and sequential data”, In Proceedings of Artificial Intelligence and Statistics, pages 299–304, San Mateo CA, 1999. Morgan Kaufman.

[3]. S. Guha, R. Rastogi, and K. Shim. “CURE: An efficient clustering algorithm for large database”. In Proceedings of the 1998 ACM SIGMOD Conference, 1998.

[4]. T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: An efficient data clustering method for very large databases. In ACM SIGMOD Conference, 1996.

[5]. K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is nearest neighbor meaningful? In ICDT Conference, 1999.

[6]. R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. Fast discovery of association rules. In Advances in Knowledge Discovery and Data Mining., pages 307–328. AAAI/MIT Press, 1996.

[7]. R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In Proceedings of 20th Conference on Very Large Databases, pages 487–499, 1994.

[8]. C. C. Aggarwal, J. L. Wolf, P. S. Yu, C. Procopiuc, and J. S. Park. “Fast algorithms for projected clustering”. In Proceedings of the 1999 ACM SIGMOD international conference on Management of data, pages 61{72. ACM Press, 1999.

[9]. R. Ng and J. Han. Efficient and effective clustering methods for spatial data mining. In Proceedings of the 20th VLDB Conference, pages 144-155, 1994.

[10]. Computational complexity between k-means and k-mediods clustering algorithms for narmal and uniform distributions of data points by T.Velmurugan and T.Santhanam.

[11]. IEE transaction Data mining and Knowledge discovery Major projected clusters in high-dimensional datasets.

K. Naveen Kumar pursuing M.Tech from St Mary’s Engineering College (JNTUH), B.Tech from SRTIST (JNTUH). His areas of interest are Advanced Computer Architecture, Data Mining & Data Warehousing.

G. Naveen Kumar, Asst Prof at SCIENT Institute of Technology, M.Tech from Satyabhama University, Chennai, B.Tech from VITS (JNTUH). His areas of interest are Computer Networks, Mining and Databases.

Ch. Veera Reddy, Assistant Professor at SCIENT Institute of Technology, M.Tech from Satyabhama University, Chennai, MSc from Pandicheri University. His areas of interest are Mobile computing, Design Analysis of Algorithms and Mining.