International Journal of Contemporary Research in Computer Science and Technology (IJCRCST) e-ISSN: 2395-5325
Volume1, Issue 9 (December '2015)

REVIEW ON CLUSTERING USING SHRINKING-BASED ALGORITHM

E.Elayaraja,
Assistant Professor,
Department of Computer Science,
Mahendra Arts and Science College,
Kallipatti, Namakkal, Tamilnadu.

K.Gopinath,
Assistant Professor,
Department of Computer Science,
Mahendra Arts and Science College,
Kallipatti, Namakkal, Tamilnadu.

Abstract: Multidimensional data has been a challenge for data analysis because of the inherent sparsely of the points. In this paper, we have present a novel data preprocessing technique called *shrinking* which optimizes the inherent characteristic of distribution of data. This data reorganization concept can be applied in many fields such as pattern recognition, data clustering and signal processing. Then, as an important application of the data shrinking preprocessing, we propose a shrinking-based approach for multi-dimensional data analysis which consists of three steps: data shrinking, cluster detection, and cluster evaluation and selection. The process of data shrinking moves data points along the direction of the density gradient, thus generating condensed, widely-separated clusters. The data-shrinking and cluster-detection steps are conducted on a sequence of grids with different cell sizes. The clusters detected at these scales are compared by a cluster-wise evaluation measurement, and the best clusters are selected as the final result. This paper shows that this approach can effectively and efficiently detect clusters in both low- and high-dimensional spaces.

Keywords: Clustering, Shrinking algorithm, data processing, multi dimensional data

I. INTRODUCTION

Multi-dimensional data has proceeded at an explosive rate in many disciplines with the advance of modern technology. Data preprocessing procedures can greatly benefit the utilization and exploration of real data. Clustering is useful for discovering groups and identifying interesting distributions in the underlying data. Data preprocessing is commonly used as a preliminary data mining practice. It transforms the data into a format that will be more easily and effectively processed for the purpose of the users. There are a number of data preprocessing techniques: data cleaning, data integration, data transformation and data reduction.

Data cleaning can be applied to remove noise and correct inconsistencies in the data. Data integration merges data from multiple sources into a coherent data store. Data transformation may improve the accuracy and efficiency of mining algorithms involving distance measurements. Data reduction can reduce the data size. These data processing techniques, when applied prior to mining, can substantially improve the overall quality of the patterns mined and/or the time required for the actual mining [1]. In this paper, we first present a data shrinking technique for preprocessing; then, we propose a cluster detection approach by finding the connected components of dense cells and a cluster evaluation approach based on the compactness of clusters.

II. CLUSTERING ANALYSIS IN MULTI DIMENSIONAL DATA

Cluster analysis is used to identify homogeneous and well-separated groups of objects in databases. The need to cluster large quantities of multi-dimensional data is widely recognized. It is a classical problem in the database, artificial intelligence, and theoretical literature, and plays an important role in many fields of business and science.

Each of the existing clustering algorithms has both advantages and disadvantages. The most common problem is rapid degeneration of performance with increasing dimensions [2], particularly with approaches originally designed for low-dimensional data. The difficulty of high-dimensional clustering is primarily due to the following characteristics of high-dimensional data:

- High-dimensional data often contain a large amount of noise (outliers). The existence of noise results in clusters which are not well-separated and degrades the effectiveness of the clustering algorithms.
- Clusters in high-dimensional spaces are commonly of various densities. Grid-based or density-based algorithms therefore have difficulty choosing a proper cell size or neighborhood radius which can find all clusters.
- Clusters in high-dimensional spaces rarely have well-defined shapes, and some algorithms assume clusters of certain shapes. For example, the algorithms in [3, 4] can efficiently find convex or spherical clusters, but they fail to detect non-spherical clusters because of their specific definition of similarity criteria.
- The effectiveness of grid-based approaches suffer when data points are clustered around a vertex of the grid and are separated in different cells, as shown in Figure 1. In the *d*-dimensional space R*d*, there may be 2*d* points distributed in this manner. The cluster formed by these points will be ignored because each of the cells covering the cluster is sparse.

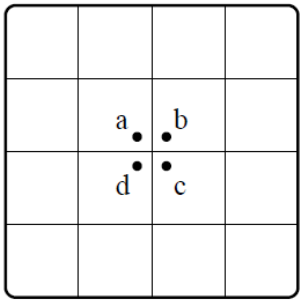

Figure 1: The four neighboring cells contain no other points.

There are also other algorithms related to data movement [5, 6], however, the existing ones are not suitable for the high dimensional data of large size because based on their definition, the time to run their process is $O(n2 * p)$ where n is the size of the input data and p is the number of iterations in the iterative process. So they are very time-consuming. A lot of approaches [7] have been proposed for evaluating the results of a clustering algorithm. Each clustering algorithm has its advantages and disadvantages. For a data set with clusters of various sizes, density, or shape, different clustering algorithms are best suited to detecting clusters of different types in the data set. No single approach combines the advantages of these various clustering algorithms while avoiding their disadvantages.

III. PROPOSED APPROACH

This novel data preprocessing technique named shrinking which optimizes the inherent characteristic of distribution of data. For a real data set, the natural data groups (if existing) it contains may be very sparse. In the data preprocessing step, if we could make data points move towards the centroid of the data groups they belong to, the natural sparse data groups will become denser, thus easier to be detected, and noises can be further isolated. Intuitively, a dense area "attracts" objects in sparse areas surrounding it, and becomes denser. Here we assume a data point is attracted to neighboring data points, and it moves towards the way the attraction is the strongest. In other words, the direction it is attracted to is determined by the distribution of its neighboring data points. Our data shrinking preprocessing computes a simulated movement of each data point in a data set that reflects its "attraction" to neighboring data points. We can also refer to the concept of infiltration mechanism [8] in which materials such as water moves from denser areas to sparser ones whereas in our case, the data point will move to a denser area nearby. When computing the attraction on a data point, those points far away from this data point can be ignored due to the little effect they impose on it. This data reorganization concept can be applied in many fields such as pattern recognition, data clustering and signal processing to facilitate a large amount of data analysis categories. Then, as an important application of the data shrinking preprocessing, we propose a shrinking based approach for multi-dimensional data analysis to address the inadequacies of current clustering algorithms in handling multi-dimensional data. This clustering method is combined with a cluster-wise evaluation measurement to select the best clusters detected at different scales.

The proposed algorithm consists of three steps which are data shrinking, cluster detection, and cluster evaluation and selection. In the data-shrinking step, data points move along the direction of the density gradient, leading to clusters which are condensed and widely-separated. Following data shrinking, clusters are detected by finding the connected components of dense cells. The data-shrinking and cluster-detection steps are grid-based. Instead of choosing a grid with a fixed cell size, we use a sequence of grids of different cell sizes. Our technique also includes a method to avoid the problem caused by points clustered near a vertex of a grid and separated in different cells, as shown in Figure 1. For each cell size, the processes of data shrinking and cluster detection are performed on two interleaved grids. Then, in the cluster evaluation and selection step, it evaluates clusters detected at different scales via a cluster-wise evaluation measurement and selects the best clusters as the final result. Although the idea of moving data points according to the density gradient has been around quite some time [9], here distinguishes mainly in the following two aspects:

- A grid-based shrinking and evaluation approach is proposed. Instead of choosing a grid with a fixed cell size, we use a sequence of increasing grid sizes to catch the cluster structures of the input data in different scales. At each scale, two grids are used; the second one is shifted diagonally from the first one.
- We integrate a compactness-based cluster evaluation into the framework. The compactness based evaluation computes both inter-cluster and intra-cluster distances (which model the inter-cluster and intra-cluster relationships, respectively) and evaluate a cluster's compactness with the ratio of the second distance to the first one.

IV. GRID SCALES FOR THE SHRINKING-BASED CLUSTERING APPROACH

To demonstrate the advantages of the data shrinking preprocessing, we applied it to the multi-dimensional clustering problem which plays an important role in many fields of business and science. We propose a grid-based approach to data shrinking and cluster detection.

Choosing grids: Grid-based clustering methods depend heavily on the proper selection of grid-cell size. Without prior knowledge of the structure of an input data set, proper grid-cell size selection is problematical. This is proposing a multiscale gridding technique to address this problem. Instead of choosing a grid with a fixed cell size, we use a sequence of grids of different cell sizes. Data shrinking and cluster detection are conducted on these grids, the detected clusters are compared, and those clusters with the best quality are selected as the final result.

Throughout this paper, we assume that the input data set X is

$$X = \{X1,X2,\ldots,Xn\};$$

Which is normalized to be within the hypercube $(0;1)^d \subset R^d$.

This apply is a simple histogram-based approach to get reasonable grid scales for the data shrinking process. We scan the input *d*-dimensional data set *X* once and get the set of histograms, one for each dimension:

$$H = \{h1,h2,\ldots,hd\}:$$

Each bin of a histogram denotes the number of data points in a certain segment on this histogram.

We set up a number b as a quantity threshold. It is used in the following algorithm to help generate *density spans*. Here we first give the definition of **density span** which will help understand the algorithm:

Definition 1: A **density span** is a combination of consecutive bins' segments on a certain dimension in which the amount of data points exceeds b. A size of a density span is the sum of the sizes of the bins it includes. For each histogram hi, i=1,...,d, we sort its bins based on the number of data points they contain in descending order. Then we start from the first bin of the ordered bin set and merge it with its neighboring bins until the total amount of data points in these bins exceeds b. At each step, we check the number of points in the bin on the left side and the one on the right side of the currently span, and choose the bin with more points in it to merge with. Thus a density span is generated as the combination of the segments of these bins. If a current span has less than b data points, but its left and right neighbors have both been assigned to a precious span already, we stop the operation on the current span and call it as an incomplete span which will not be considered in the following procedure of generating multiple grid scales. The operation is continued until all the non-empty bins of this histogram is in some density spans or some incomplete spans. Each histogram has a set of density spans.

First we sort the sizes by ascending order. Then starting from the smallest size so, we include the current size into cluster T1 until we come across a size s0 such that s0¡so > so *5%. Then we start from s0 and do the same procedure to get cluster T2, and so on. For each cluster Ti, we denote the number of sizes in it as Ni, and denote the average value of the sizes in it as Si. We sort Si based on Ni by descending order and choose first Ks ones as the multiple scales for the following data shrinking and cluster detection procedures. In other words, those sizes of density spans which appear often are chosen. Algorithm 1 describes the procedure of the density span generation on a certain dimension. The value b depends on the size of the input data set X. Normally it can be set as a certain percentage of the number of data points in X. There is a balance in choosing a value for Ks: smaller Ks can increase the precision of cluster detection, while larger Ks can save time. The time complexity for this method is determined by the dimensionality d of X and the amount of bins Bn in each histogram. The time required to perform Algorithm 1 is O(Bn logBn).

Algorithm 1 (Density span generation)
Input: histogram *hi*
Output: Density span set of *hi*
1) Sort the bins of *hi* in the descending order;
2) Beginning from the first bin of the ordered bin set, merge it with its neighbors until the total amount of data points included exceeds b;
3) Repeat step 2 until all non-empty bins are included in some density spans or some incomplete spans;
4) Output the density span set.

The multiscale gridding scheme proposed above not only facilitates the determination of a proper cell size but also offers advantages for handling data sets with clusters of various densities. The grid with a smaller cell size (shown in solid lines) can distinguish the left two clusters but fails to detect the right cluster, while the converse is true for the grid with a larger cell size (shown in dashed lines). For data sets of this kind, a multiscale gridding method is needed to distinguish all clusters.

V. DATA SHRINKING

In data shrinking, each data point moves along the direction of the density gradient and the data set shrinks toward the inside of the clusters. Points are "attracted" by their neighbors and move to create denser clusters. This process is repeated until the data are stabilized or the number of iterations exceeds a threshold. The neighboring relationship of the points in the data set is grid-based. The space is first subdivided into grid cells. Points in sparse cells are considered to be noise or outliers and will be ignored in the data-shrinking process. Assume a dense cell C with neighboring cells surrounding C. Data shrinking proceeds iteratively; in each iteration, points in the dense cells move toward the data centroid of the neighboring cells. The iterations terminate if the average movement of all points is less than a threshold or if the number of iterations exceeds a threshold. The major motivation for ignoring sparse cells is computation time. If the grid cells are small, the number of non-empty cells can be O (n), where n is the number of data points. The computation of data movement for all non-empty cells takes a length of time quadratic to the number of non-empty cells, which is O (n2). By ignoring sparse cells in the data movement, dramatic time savings can be realized.

VI.CLUSTER EVALUATION AND SELECTION

After the cluster detection step, we evaluate the clustering results. There are several ways to define what is a good clustering ([11] .etc). Most conventional clustering validity measurements evaluate clustering algorithms by measuring

the overall quality of the clusters. However, each clustering algorithm has its advantages and disadvantages. For a data set with clusters of various sizes, densities, or shapes, different clustering algorithms are best suited to detecting the clusters of different types in the data set. No single approach combines the advantages of the various clustering algorithms while avoiding their disadvantages. Here, introduce a cluster-wise measurement which provides an evaluation method for individual clusters.

A cluster in a data set is a subset in which the included points have a closer relationship to each other than to points outside the cluster. In the literature [12, 13], the intra-cluster relationship is measured by compactness and the inter-cluster relationship is measured by separation. Compactness is a relative term; an object is compact in comparison to a looser surrounding environment. We use the term compactness to measure the quality of a cluster on the basis of intra-cluster and inter-cluster relationships. This definition of compactness is used to evaluate clusters detected at different scales and to then select the best clusters as the final result.

VII.CONCLUSION

In this paper, we proposed a new data preprocessing technique called *shrinking* which optimizes the inherent characteristic of distribution of datas. We applied the technique and proposed a novel data analysis method which consists of three steps: data shrinking, cluster detection, and cluster evaluation and selection. The methods can be effectively and efficiently detect clusters of various densities or shapes in a noisy data set of any dimensions. The data-shrinking process still poses many open issues. The shrinking process as applied to a data set of well-formed shape is a repeated process which transforms the data set into a shape with no boundary. However, most real-world, high-dimensional data sets do not have well-defined shapes. It is therefore of both theoretical and practical interest to fully understand how the shape of a real data set is transformed during the shrinking process. This understanding would provide insights into the geometrical and topological properties of high-dimensional data sets.

REFERENCES

[1] C.C. Aggarwal and P. Yu. Finding generalized projected clusters in high dimensional spaces. In Proceedings of the ACM SIGMOD CONFERENCE on Management of Data, pages 70–81, Dallas, Texas, 2000.

[2] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. On the surprising behavior of distance metrics in high dimensional space. Lecture Notes in Computer Science, 1973, 2001.

[3] Charu C. Aggarwal, C. Procopiuc, J.L. Wolf, P. Yu, and J.S. Park. Fast algorithms for projected clustering. In Proceedings of the ACM SIGMOD CONFERENCE on Management of Data, pages 61–72, Philadelphia, PA, 1999.

[4] Alexander Hinneburg, Charu C. Aggarwal, and Daniel A. Keim. What is the nearest neighbor in high dimensional spaces? In The VLDB Journal, pages 506–515, 2000.

[5] Alexander Hinneburg and Daniel A.Keim. An efficient approach to clustering in large multimedia databases with noise. In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, pages 58–65, New York, August 1998.

[6] J. MacQueen. Some methods for classification and analysis of multivariate observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. Volume I,Statistics., 1967.

[7] Jagadish, H.V., Madar, J. and Ng, R. Semantic compression and pattern extraction with fasicicles. In VLDB, pages 186–196, 1999.

[8] A. K. Jain and R. C. Dubes. Algorithms for Clustering Data. Prentice Hall, 1988.

[9] A.K. Jain, M.N. Murty, and P.J. Flyn. Data clustering: A review. ACM Computing Surveys, 31(3), 1999.

[10] Jiawei Han, Micheline Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, 2001.

[11] K. Fukunaga. Introduction to Statistical Pattern Recognition. 1990.

[12] K.V. Ravi Kanth, Divyakant Agrawal, and Ambuj Singh. Dimensionality reduction for similarity searching in dynamic databases. In Proceedings of the ACM SIGMOD CONFERENCE on Management of Data, pages 166–176, Seattle, WA, 1998.

[13] L. Kaufman and P. J. Rousseeuw. Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley & Sons, 1990.

[14] Liu, J. and Wang, W. OP-Cluster: Clustering by Tendency in High Dimensional Space. Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03), November 19-22 2003.

[15] Maria Halkidi, Michalis Vazirgiannis. Clustering Validity Assessment: Finding the Optimal Partitioning of a Data Set. In ICDM, 2001.

