International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 12 (2017) pp. 3487-3490

Scalable Hyperspace Partitioning Based Data Preprocessing Algorithm for Distance-Metric Based Clustering in Data Mining

Manju Pandey
Assistant Professor, Department of Computer Applications, National Institute of Technology, Raipur 492010, Chhattisgarh, India.

Ravi K Jade
Assistant Professor, Department of Mining Engineering, National Institute of Technology, Raipur 492010, Chhattisgarh, India.

[1]ORCID: 0000-0002-5817-4121

Abstract

In the present paper, partitioning of the n-dimensional hyperspace into n-dimensional hypercubes, and determination of the point density of the hypercubes is demonstrated to be an effective and scalable data preprocessing technique to improve the accuracy and performance of distance-metric based clustering algorithms in data mining. In the paper we have considered 2-dimensional space for the relative ease of illustration and demonstrated the accuracy and performance improvements resulting from this preprocessing technique for the K-means algorithm based on the Euclidean norm distance-metric.

Keywords: data mining, clustering, data preprocessing, partitioning, hyperspace, hypercubes, n-dimensional space, n-dimensional cubes, n-dimensional vectors, distance-metric, Euclidean norm, k-means algorithm

INTRODUCTION

Clustering is an important topic in the field of data mining and has numerous real-world applications. Applications include market baskets, geographical data, insurance, city-planning, earthquake studies, image processing, and character recognition, besides others. At its core distance-metric clustering is about determination of appropriate clusters or groups of close n-dimensional data points, also called point clouds, in n-dimensional space where the relative closeness of the data points is a function of the distance-metric applied in the specific case. Widely applied distance-metric based clustering algorithms including the K-means algorithm and the Fuzzy C-means algorithm are approximate algorithms whose results can be improved by pre-processing the raw data used for clustering. In the paper partitioning of the n-dimensional hyperspace into n-dimensional hypercubes and determination of the point density of the resulting hypercubes is proposed as a trivially parallel scalable data pre-processing method to improve the accuracy and performance of distance-metric based clustering algorithms in data mining. While the concepts developed in this paper are applicable to n-dimensional hyperspaces, the paper considers the 2-dimensional case for the relative ease of illustration. The distance-metric based K-means algorithm is taken up for the present study and it is shown how the partitioning based preprocessing works in the case of the K-means algorithm in 2-dimensional space to improve the clustering results.

A recursive partitioning algorithm has been applied to problems involving spatial clustering (1) whereas hierarchical partitioning has been described in (2). In (3) and (4) dynamic adjustment of the partition size has been done. A survey of grid based clustering algorithms has been done in (5). The grid algorithms partition the space into rectangular regions and carry out the clustering algorithms on the resulting partitions. Applications to the analysis and visualization of web opinion development and social interactions are described in (6). The application of topographic maps to density based clustering has been described in (7) whereas the application of level sets to density based clustering has been discussed in (8). In (9), a multi-resolution clustering based approach has been described for very large spatial databases. The application of grid algorithms to data stream mining has been described in (10-12) and its application to wireless sensor networks has been described in (13). Application of the algorithm to high dimensional data and to very large datasets is described in (14) and (15) respectively.

Data Set:

Artificial 2-d Data for pre-processing is generated by the Synthetic Data Generator discussed in (1).

Organization of the Paper

This paper starts with a brief overview of clustering. The K-means clustering algorithm and its weaknesses are briefly discussed next. This is followed by a generalized discussion of the proposed hyperspace partitioning and hypercube point density data preprocessing method for the case of n-dimensional space. Following this is a discussion of the method as applied to the case of 2-dimensional data points in 2-dimensional space. The next section details how the proposed pre-processing method improves the accuracy and performance of the K-means clustering algorithm. Following this we discuss the results. The last section deals with conclusions and future work.

Clustering

At an abstract level there is a set of points in a high dimensional space. It is required to determine if there are distinct point clouds, groups or "clusters," that is, subsets of points that are close to each other and far away from points in other groups. A clustering method should decide if there indeed are clusters and, if yes, assign a cluster label to each point. In distance-metric based clustering algorithms, the distance measure for determining the relative closeness of the points in high dimensional space is an appropriate distance-metric. Commonly applied distance metrics include the Euclidean-norm distance metric, the Manhattan distance, etc.

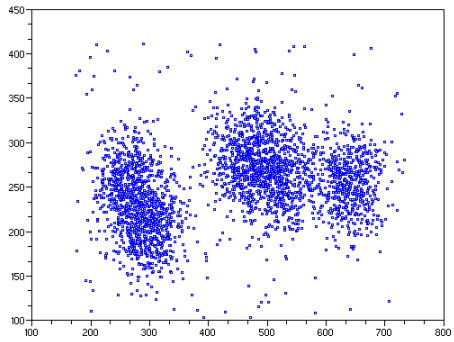

Figure 1: Dataset with Four Point Clusters/Clouds

The Euclidean distance between two points (x_1, x_2) and (y_1, y_2) in 2-dimensional Euclidean metric space is given by the following Pythagorean formula

$$d = ((y_1 - x_1)^2 + (y_2 - x_2)^2)^{\frac{1}{2}} \quad (1)$$

The equivalent Pythagorean formula for the distance between the two points (x_1, x_2, …, x_n) and (y_1, y_2,…, y_n) in n-dimensional Euclidean metric space is as follows:

$$d = (\sum (y_i - x_i))^{\frac{1}{2}} \quad i=1,2,...,n \quad (2)$$

Here i=1,2,…,n are the n-dimensions of space.

The K-means clustering Algorithm

To determine groups/clouds/clusters of closely spaced points in a dataset, the K-means algorithm proceeds step-wise as follows:

1. In the space containing the dataset points that are to be clustered, K points are chosen as the initial cluster centroids.
2. Distance of each dataset point is computed relative to all the cluster centroids, and the dataset point is assigned to the cluster whose centroid is the nearest.

   Pythagorean formula for distance between the two points in n-dimensional Euclidean metric space of the forms given in (1) and (2) are used.
3. Once all the data points have been so assigned, the positions of the K centroids are recomputed.

   The formulas used are:

$$x_{new} = \frac{\sum x_j}{p} \quad j=1,2,\ldots,p \quad (3)$$

$$y_{new} = \frac{\sum y_j}{p} \quad j=1,2,\ldots,p \quad (4)$$

   Here j=1,2,…,p are the p data points of the cluster whose centroid is being recomputed.

4. Steps 2 and 3 are repeated until the centroids become more or less static and do not change appreciably with subsequent iterations.

In this manner, data points are separated into groups from which the metric to be minimized can be calculated.

To summarize, the goal of this algorithm is to minimize a squared error objective function:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 \quad (5)$$

Here $\| x_i^{(j)} - c_j \|^2$ is the distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j.

J is a measure of the distance of the *n* data points from their respective cluster centres.

The K-means algorithm may be viewed as a greedy algorithm for partitioning the n samples into k clusters so as to minimize the sum of the squared distances to the cluster centers.

The K-means algorithm has following five known weaknesses:

1. The method to initialize the cluster centroids has not been specified. A commonly applied technique to start is to randomly choose the k initial cluster centroids.
2. The final results are dependent on the initial cluster centroids, which determine the initial means, and as a result of this, many times suboptimal partitions are found. The solution is to try a number of different starting points.
3. The set of samples closest to $\mathbf{c}_j$ may be empty, so that $\mathbf{c}_j$ cannot be updated.
4. Results depend on the distance-metric which is used to measure the distance $\| \mathbf{x} - \mathbf{c}_j \|$. One solution, which is not desirable in all cases, is to normalize each variable by its standard deviation.
5. The results depend on the value of k.

Hyperspace Partitioning for Data Clustering

In the first step, the bounds of the data points of the dataset in each dimension are determined to enable the narrowing down of the search space. As the name suggests, hyperspace partitioning is a data pre-processing method in which each dimension of the imaginary n-dimensional hyperspace which contains all points of the dataset is first partitioned or divided by equispaced imaginary hyperplanes on which the dimension being partitioned assumes a constant value. After this second step, the n-dimensional space has been partitioned into imaginary hyperboxes whose size along each dimension depends on the spacing of the imaginary partitioning hyperplanes in that direction. If the partitioning is equispaced in all dimensions, then we have a hyperspace partitioned into imaginary hypercuboids of the same size in all dimensions. The point density of the imaginary hyperbox or hypercube is determined in the third step. For this each of the dimensions of all the points are compared to determine in exactly which hyperbox or hypercube, each of the points is present. This gives the point density of each hypercube in the hyperspace. The hypercubes are arranged in descending order of point density. This data is then used for determining good regions for initial cluster centroids, as well as the number of clusters K. Consequently, we are able to address the weaknesses of the K-means clustering algorithm. First, based on this data, we have a good first estimate of the initial cluster centroids. Second, a good choice of initial cluster centroids improves the initial means, and therefore the final results. The problem of empty sample sets close to a particular centroid is also solved. On the basis of this data, It is also possible to have a good first hand estimate of the number of initial clusters K.

Space Partitioning Applied to Data Points in 2-D space

Figure 2 shows a dataset in 2-D space which has been partitioned into boxes which have size of 100 units along x-dimension and 50 units along y-dimension. The dataset consists of 3000 data points. The mean number of points in the boxes shown is ~60.

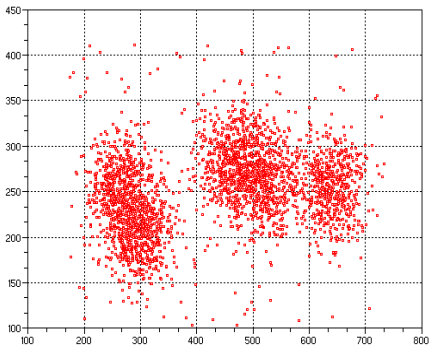

Figure 2: Dataset with Three Point Clusters/Clouds with 2-D space partitioned into boxes

Figure 3 shows the number density distribution of the data points in the 2-dimensional space. The concentration of the data points can be easily seen from this figure.

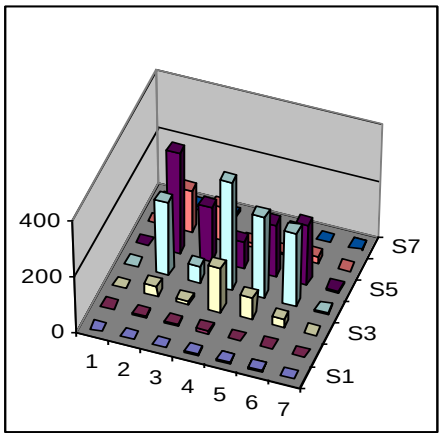

Figure 3: Number Density of Points in 2-D Space

From the figure it is also possible to figure out good initial centroids for the K-means algorithm, as well as other algorithms which are based on distance-metrics and choice of initial cluster centroids. It is also possible to determine the value of K, the number of clusters into which the data set may be partitioned. The methods discussed so far require manual

intervention at this stage and are mainly based on visualization. The pertinent question to ask at this juncture is the possible automation of these methods. This assumes particular significance since manual intervention increases the turnaround time. Furthermore, it is the case that a large set of data clustering problems in the field of data mining typically deal with data sets having data vectors with more than 3-dimensions. Visualization of data sets becomes next-to-impossible in the case of higher dimensional data and appropriate computer methods must be developed for determining regions of higher dimensional space with greater density of points. Apart from this the visualization of large data sets where data points are large in number and scattered present difficulties even for the case of 2-D data sets. Even though the data set may be 2-dimensional, multiple scales of resolution also pose difficulties in representation and proper visualization. Therefore, there is clear potential for automated methods to determine regions of higher dimensional space with a greater number density of data set points.

REFERENCES

[1] Jiang X, Cooper GF. A Recursive Algorithm for Spatial Cluster Detection. In AMIA Annual Symposium; 2007; Chicago. p. 369-373.

[2] Wang W, Yang X, Muntz R. STING: A Statistical Information Grid Approach to Spatial Data Mining. In 23rd VLDB Conference; 1997; Athens. p. 186-195.

[3] Liao WK, Liu Y, Choudhary A. A Grid Based Clustering Algorithm using Adaptive Mesh Refinement", 7th Workshop on , 2004. In 7th Mining Scientific and Engineering Datasets; 2004; Florida. p. 1-9.

[4] Lin Np, chang CI, Jan NY, Chen HJ, Hao HW. A Deflected Grid-based Algorithm for Clustering Analysis. International Journal of Mathematical Models and Methods in Applied Sciences. 2008 March; 7(3).

[5] Ilango MR, Mohan V. A Survey of Grid Based Clustering Algorithms", , Vol. 2(8), 2010, 3441-3446. International Journal of Engineering Science and Technology. 2010; 2(8).

[6] Yang CC, Ng TD. Analyzing and visualizing web Opinion development and social Interactions with Density-Based Clustering" pp 1144-1155 vol(41),2011. IEEE Transactions on Systems, Man, and Cybernetics. 2011; 41.

[7] Hulle MV. M. Van Hulle,"Density-based clustering with topographic maps", ,pp:204-207 vol(10), 1999. IEEE Transactions on Neutral Networks. 1999; 10.

[8] Wang XF, Huang SD. A novel Density-Based clustering Framework by using Level Set Method. IEEE Transactions on Knowledge and Data Engineering. 2009; 21.

[9] Sheikholeslami G, Chatterjee S, Zhang A. WaveCluster: A Multi-Resolution Clustering Approach for Very Large Spatial Databases. In 24th VLDB Conference; 1998; New York. p. 428-439.

[10] Aggarwal CC, Hang J, Wang J, Yu PS. A framework for clustering evolving data streams. In VLDB; 2003; Berlin. p. 81-92.

[11] Wei-Heng JZ, Jian Y, Yi-Huang X. Arbitrary Shape Cluster Algorithm for Clustering Data Stream. Journal of Software. 2006; 17(3).

[12] Qing-Bao L, Chao-Fan D, Su D, Wei-Ming Z. Grid-based Data Stream Clustering Algorithm. Science of computer. 2007; 34(3).

[13] Xu Z, Yin Y, Wang J. A Density based Energy efficient Clustering Algorithm for Wireless Sensor Networks. International Journal of Future Generation Communication and Networking. 2013 Febuary; 6(1).

[14] Agarwal R, Gehrke J, Gunopulos D, Raghavan P. Automatic Subspace Clustering of High Dimensional Data. Data Mining and Knowledge Discovery. 2005; 11.

[15] Goil S, Nagesh H, Choudhary A. MAFIA: Efficient and Scalable Clustering for very large data sets. Technical. Illinois: Northwestern University, Center for Parallel and Distributed Computing; 1999. Report No.: CPDC – TR – 9906 – 010.