Scientific Research

A Density Clustering Algorithm Based on Data Partitioning

Dongping LI

Kunming University, Kunming, China

Email: R3mj8@21cn.com, yfldp@sina.com

Abstract: As a density clustering algorithm, DBSCAN can find the denser part of data-centered samples, and generalize the category in which sample is relatively centered. This article analyzes the traditional DBSCAN clustering algorithm and its flaw, and discusses an implementation of a density clustering algorithm based on data partitioning. The algorithm can solve the memory support and I/O consuming problems when it handles high volume data, and gets faster clustering speed and the best clustering effect.

Keywords: cluster analysis, DBSCAN algorithm, RDBSCAN algorithm, data partitioning

1 Introduction

Clustering Analysis is always an important research topic of data mining field. Clustering is a procedure that groups the physical or abstract object sets into several categories or clusters which is composed with the similar objects. The clusters grouped by clustering are several non-empty sets. The objects in a cluster have the most similarity, and the objects in different clusters have the most dissimilarity. As an independent analysis tool of data mining, clustering analysis not only can obtain the status of the distribution of data, observe the characteristics of every cluster of data and obtain the analysis result, but is often used as the preprocess of the other data mining analysis methods in many practical applications. At present, it has five clustering algorithms: the method based on distance, the method based on density, the method based on hierarchy, the method based on model and the method based on grid [1].

Based on the foregoing Density Based Spatial Clustering of Applications with Noise (DBSCAN), this article introduces an algorithm of density clustering based on data partitioning (RDBSCAN), it solves the memory support and I/O consuming problems resulted from the large volume data when building R* trees, and avoids the possible incorrect result due to adopt the globe Eps, and at last implements RDBSCAN algorithm on simulated data. It verifies the effectiveness and feasibility of data partitioning strategy.

2 DASCAN Algorithm

DBSCAN Algorithm was introduced by Martin Ester et.al. in 1996. The algorithm groups the high density area to a cluster. There are two important parameters in the algorithm: neighbor radius of object Eps and the minimum object number in neighbor MinPts. The target of DBSCAN algorithm is that find out the sets of density-joined objects, in other word, find out the different clusters. In DBSCAN, cluster means the high density object area divided by the low density object area in data space. The basic idea of DBSCAN is: In order to find out a density-joined area, clustering from a random object p. If p is kernel object, that is p is the center of a circle, Eps is radius and the number of objects in the circle equals or great than MinPts, then the algorithm returns a density-joined set, identifies all objects in this set as in the same cluster; if p is not a kernel object, there is no other object reachable from p density, then p is identified as noise. DBSCAN algorithm handle every un-scanned point according to the above, at last the density-joined objects are grouped to the same cluster, and the object that is not belong to any cluster is noise. For every kernel object in data sets, the density-joined set can be returned [2].

The outstanding characteristic of DBSCAN algorithm is that it can find out any arbitrary shape cluster, it breaks the limitation of other algorithms which only can find out the fixed shape cluster. It is not sensible to noise, so it enormously improves the flexibility of clustering algorithm, and the clustering result is un-acted on noise. At the same time, we can also discover easily:

1) The whole algorithm needs the memory big enough. When searching all neighbors of kernel point p related to Eps and MinPts, it needs build R* tree index of all whole data space. The memory is the bottleneck restricting the algorithm when data space expands sharply.

2) Determination of Eps parameter is related to more artificial factors, there is no scientific guidance standard. DBSCAN algorithm usually uses a globe Eps to cluster data space. In the circumstance of data distribution is uneven in data space, that is the circumstance of the densities is largely different between clusters, the disadvantage is obvious.

3 Improvement of the Algorithm

Aim at the flaws of DBSCAN algorithm, the article introduces such an improved strategy: By scanning data set once or more times, some approximate distribution of

978-1-935068-10-5

the whole input data set is obtained, and the whole data set is divided into several partitions according to the approximate distribution. The different Eps are determined in different partition, and apply DBSCAN algorithm in different partitions using the Eps respectively until the most data point is identified to a cluster. By this way, which is clustering in different partition, the whole clustering procedure is not blind, and improve the result. This is the density clustering algorithm based on data partitioning (RDBSCAN).

3.1 RDBSCAN Algorithm Procedure

Input: whole data space.

Output: the clustering resulting sets and noise information.

1) By scanning the whole data set, the density distribution of data set is obtained. The whole data space is divided into several sub area according to density distribution, and the partition results is put in the set D.

2) Apply DBSCAN for the member of D one by one (record the border point and noise point of every cluster), until a majority of points is belong to a cluster.

3) Combine the clusters generated in various partitions, denoise, and output the last result.

It is make out by this that the implementation of RDBSCAN algorithm is mainly divided into three parts: data partitioning, applying DBSCAN algorithm and combining clustering results.

3.2 Data Partitioning

In RDBSCAN algorithm, the core problem is that data partition is determinated according to the density distribution of data set. Analyzing the data partitioning strategies referred to the articles [3,4], we noticed that they all use histogram to analyze data space distribution. It is obvious that the statistical function of histogram is capable of analyzing data space distribution.

This article adopst an simplified histogram. Its procedure is similar to the drawing method of histogram in statistics. The difference is that its vertical coordinate is the number of points which is located in various partitions. However, the horizontal coordinate is the same to the histogram in statistics. For instance:

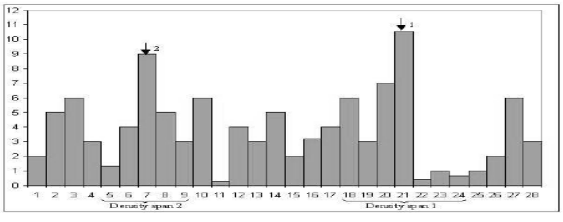

Figure 1. Histogram Density Distribution

As shown in Figure 1. Every rectangle denotes the number of data points which is belong to a partition at the dimension in data space. The Rectangle 1 means there are 11 points between 20 and 21 at the dimension distributed in data space. Rectangle 2 means there are 9 points at the dimension distributed in data space.

The density distribution is analyzed out according to this histogram. In figure 1, there are 28 rectangles represent 28 different partition at the dimension. We sort the rectangles according to the number of data points which is in the rectangle, and put in the set A in descending order. The rectangle 1 and 2 in the figure are the first two rectangles which include the most data points, respectively represent the two partitions with the most data points at the dimension.

Then, we start from the rectangle 1 with the most data points, combine it with its neighbor rectangles, until the data point of the combined rectangle excesses a threshold X (It is generally a percent of input data number). The new rectangle is labeled as S1 and the span of the new rectangle is recorded as Wide, which is the number of the small rectangles which is combiner into the new rectangle.

This combining method guarantees there are points more enough in the area of partition and the space which has the most points will be a candidate. The reason of combining is to avoid the partition will not be divided more fragmentally. More fragmental, more computational complexity the following combining of clustering partition has. Thus, the algorithm is encumbered by the time used by combining. The discussion about the value of X is, if X is too large, every partition will be large, then the overhead of I/O is not much difference before unimproved, and it is meaningless; if X is too small, every partition will be fragmental and like the combining, the system overhead increases, so its value usually is between 30%–70%.

Moreover, for the Rectangle 2 in A, repeat the procedure the above, S2 is obtained. Repeat like this until all rectangles is belong to a bigger rectangles Si, i=1, 2,…,1/X, 1/X is the number of the new rectangle in the histogram.

At last, to find out the largest density distribution area at the dimension, compute di of every rectangle Si in the new generated rectangle set. di is defined as follows,

Forum: di=Pointnumber /Wi (i=1, 2,…,1/x)

The Pointnumber represents the point number in the rectangle. The Wi represents the number of the small rectangle combined in the new rectangle. The di is the density of the partition; the corresponding result is kept in the set D.

Until now, the analysis of data histogram at the first dimension is over. The data density distribution at the first dimension is kept in D. We just find out the largest di then we know the largest density distributed partition that is di represent.

We know the point distribution at the dimension in data set after analyzed the data at a dimension.

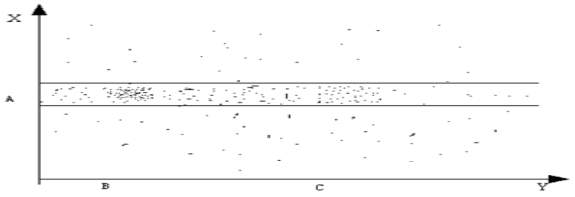

Figure 2. Density Distribution at the first Dimension

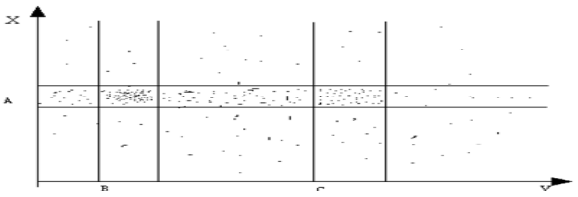

Figure 3. Density Distribution at the second Dimension

Then we find out the largest density section data point (A) at the dimension, analyze the data distribution at the other dimension using the same method. The following is that the largest density area will be found out at the second dimension.

After scanned at X dimension, the data density of area A is found out to be largest. Then continue to scan density distribution at Y dimension to all the data point in the area A. It is found out that the points in Area A and in Area B at the Y dimension has largest density. Then we can educe that in area A, the density of data points of the area which has a common boundary between A and B is the largest. By the procedure, we finally determinate the area which has a common boundary between A and B in the area A is the largest density area.

Then we continue find out the second largest density area C in the area A at the Y dimension, until all points in the area A are belong to an area. Till here, it is over for the area A to be analyzed for the density distribution. Then repeat the above procedure for the other area at the X dimension (the other area is kept in the set D according to the density).

By this way, we divided the whole data space into many partitions according to density distribution. The density of data points of every partition is different. Adopting the different Eps value, the two flaws of traditional DBSCAN algorithm is avoided. This way can be extended to higher dimension space.

3.3 Combining Partition Clustering Result

After clustering every partition, it is needed to combine the partition clustering result. The article [5] introduced a better solution, and the experimental result of the article indicates the algorithm is correct and effective. This article here adopts its solution.

Three conditions are needed to be considered.

1) Consider the same cluster can be divided into different partitions, so it is necessary to combine the partitions.

Cluster A and cluster B is to be combined if and only if:

① A and B is located in the two adjacent partitions PA and PB

② suppose Eps(PA), Eps (PB) is respectively the Eps value of PA and PB. Eps (PA, PB)=min{Eps (PA) ， Eps（PB)}, EA , EB is border area point set which is saved during the local clustering procedure. For ∃p∈ EA, ∃q∈EB,satisfy DISTANCE{p,q}≤Eps(PA,PB)。

2) The point labels as noise in every partition is possible the border point of a cluster in an adjacent area. It is necessary to be identified as noise. A noise point Pn is identified as a cluster C if and only if:

① Pn and C are located in two adjacent paritions respectively.

② ∃ p∈EC，EC is border area point set which is saved dring local clustering procedure ， satisfy DISTANCE {pn ， p }≤Eps(PC)。

3) The other condition is the noise can be in a new cluster with the other noise points in adjacent area.

Suppose PA and PB are two adjacent partitions, EPA and EPB are border area point set of these two partitions. Consider the noise point set of EPA and EPB as SN, if ∃p0∈SN, ∃pi∈SN (i= 1, 2, ⋯,m , m ≥Minpts) , and pi≠p0. Satisfy DISTANCE{p0, pi}≤Eps (PA, PB) ， Then consider p0 is a kernel point of a new global cluster，{pi}(i= 1, 2,…,m)is the other poins in the new cluster. The rest noise point will be handled according to the procedure (2).

4 Experimental Analysis

In a two dimension space, by handling the simulated data in Figure 4, the effectiveness of the density cluster acquired method based on data partitioning can be analyzed and verified, and compared to the last clustering center which is obtained by DBSCAN algorithm. The parameter used by the density clustering center acquired method is: the number of the area which histogram divided is 60; the threshold that is to combine areas is 35%. DBSCAN algorithm Eps: 26, MinPts: 112;

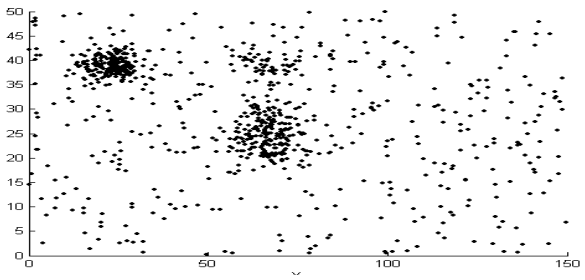

Figure 4. Simulated data

978-1-935068-10-5 © 2010 SciRes.

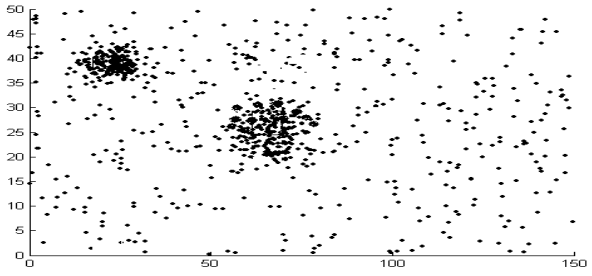

Figure 5. DBSCAN algorithm

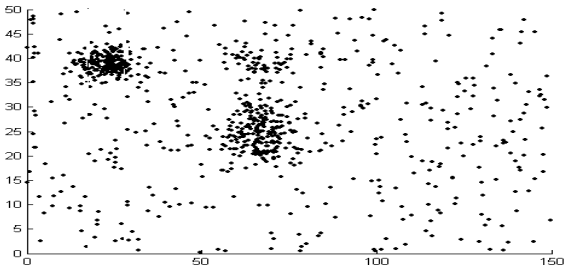

Figure 6. RDBSCAN algorithm

We can see from the results, the clustering result of the density clustering based on data partitioning is the best.

Compared to the efficiency of the two algorithms, the computational complexity is O (nlogn). Because RDBSCAN algorithm uses the strategy that apply DBSCAN algorithm in data partitions, then when DBSCAN algorithm is applied in every data partition, due to the less data point number, the needed space used by RDBSCAN algorithm in every data partition must be less than DBSCAN. When the data volume increase sharply, the advantage of data partitioning algorithm which need the less memory is adequately combined out.

So, the space adaptability of RDBSCAN algorithm is obvious better than DBSCAN algorithm. Meanwhile the clustering result of RDBSCAN algorithm can solve the problem of data distributed unevenly which is not solved by DBSCAN algorithm. So RDBSCAN algorithm is much better than DBSCAN algorithm.

5 Ending

DBSCAN algorithm can find out any shaped cluster in data set with noise, but when data volume is huge, it needs huge memory and I/O overhead. When Data distributed is uneven, the effect of clustering is worse because it use uniform globe variable. I introduced the RDBSCAN algorithm to solve the problem of DBSCAN. The algorithm divided data set into many local parts. It uses different parameter values when clustering every local part, and combines the results of local parts at last. Experimental indicates RDBSCAN algorithm is better than original DBSCAN algorithm in clustering quality.

References

[1] Jiawei Han, Micheline Kamber, data mining concepts and techniques [M], Second Edition, Beijing, Machinery Industry Press, 2007 P128-P140.

[2] Sander J, Ester M, K riegel H P *et al*. Density based clustering inspatial databases: the algorithm GDBSCAN and its applications [J]. Data Mining and Knowledge Discovery, Kluwer Academic Publishers, 1998, 2(2):487-493.

[3] Zhou Shui-ying, Zhou Ao-ying, aData Partitioning Based DBSCAN Algorithm, Journal of computer research & development, Vo1.37. No.1 0 Oct.2000, 1153-1159.

[4] Yong Shi, Yuqing Song and Aidong Zhang a Shrinking-Based Approach for Multi-Dimensional DataAnalysis: Department of Computer Science and Engineering State University of New York at Buffalo Buffalo, NY 14260.

[5] Vitter J. Random sampling w ith reservo ir. ACM Trans on Mathematical Software, 1985, 11 (1): 37-57.