An Efficient Clustering Algorithm for 2D Multi-density Dataset in Large Database

Ying Xia[1,2], GuoYin Wang[2], Song Gao[2]
[1]*School of Information Science and Technology, Southwest Jiaotong University, Chengdu, 600031, P.R.China*
[2]*School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Nan'an Distinct ChongQing, 400065, P.R.China*
xiaying@cqupt.edu.cn, wanggy@cqupt.edu.cn, gao_fly@hotmail.com

Abstract

Spatial clustering is an important component of spatial data mining. The requirement of detecting clusters of points arises in many applications. One of the challenges in spatial clustering is to find clusters on multi-density dataset. In this paper, a Grid-based Density-Confidence-Interval Clustering algorithm for 2-dimensional multi-density dataset is proposed, called GDCIC. The proposed algorithm combines the density confidence interval with grid-based clustering, and produces accurate density estimation in local areas for local density thresholds. Local dense areas are distinguished from sparse areas or outliers according to these thresholds. Experiments based on both synthetic and real datasets verify that the algorithm is efficiently for multi-data sets and handle outliers effectively.

1. Introduction

Data mining is the process of discovering interesting and potentially useful patterns of information embedded in large databases. Spatial data mining is a niche area within data mining for the rapid analysis of spatial data [1]. It aims to automate the process of understanding spatial data by representing the data in a concise manner and reorganizing spatial databases to accommodate data semantics.

Spatial clustering is to group similar objects based on their distance, connectivity, or their relative density in space. The clustering algorithms can be categorized into four categories: partitioning, hierarchical, density-based and grid-based clustering algorithm [1], [2]. Grid-based clustering algorithms first quantize the clustering space into a finite number of cells and then perform the required operations on the quantized space. Cells that contain more than certain number of points are treated as dense. The dense cells are connected to form the clusters. The main advantage is its fast processing time especially for large dataset, which is only dependent on the number of cells in the quantized space. It's also order-insensitive for the input data points. Considering the advantages mentioned above, we adopt grid-based method in GDCIC for large dataset in spatial database.

One of the main applications of clustering to spatial databases is to find clusters of points (where points represent the spatial objects) which are physically close together in 2-dimension geographic spaces. There is one case of datasets with multi-density clusters. In this case, small clusters with small number of points in a local area are possible to be missed by a global density threshold. Several algorithms are proposed to solve this problem, such as indexed-based density biased sampling, Chameleon, SNN etc. Indexed-based density biased sampling [3] can avoid losing small clusters by selecting enough samples in local dense areas, but the result is not accurate as the influence of samples. Chameleon [4] can handle multi-density datasets, but its time complexity is too high on large dataset. Although SNN [5] can find clusters of varying shapes, sizes and densities, the disadvantage of SNN is that the degree of precision is low on the multi-density clustering and finding outliers. Some previous works are done based on density-grid based clustering algorithm, such as DGCL [6], but a uniform density threshold is used which might cause low density clusters lost.

In this paper, we present an effective Grid-based Density-Confidence-Interval Clustering algorithm (GDCIC). By using the technique of density confidence interval, a local density threshold is calculated to distinguish local dense areas from sparse areas or outliers in a quantized space. The experiment

2007 International Conference on Multimedia and Ubiquitous Engineering(MUE'07)
0-7695-2777-9/07 $20.00 © 2007 IEEE

shows that it has higher performance than the methods we mentioned above for handling multi-datasets.

The rest of this paper is organized as follows. Section 2 introduces some definitions used in GDCIC. The technique of density-confidence-interval is described in section 3. Section 4 presents the algorithm GDCIC which discovers multi-density clusters. Section 5 analyses the time complexity. In section 6, we show the experiment results on both synthetic and real datasets. A conclusion is presented in the last section.

2. Some Definitions in GDCIC

In the following, we introduce some definitions which are used in GDCIC.

Definition 1 (Confidence interval) A confidence interval gives an estimated range of values which is likely to include an unknown population parameter, the estimated range being calculated from a given set of sample data. For example, the 95% confidence interval is constructed in such a way that 95% of such intervals will contain the parameter.

Definition 2 (Cell density) The cell density is the number of point spatial objects which are contained in one cell.

Definition 3 (Useful cells) The grid cells that contain data points are treated as the candidates of useful cells. During the process of clustering, some useful cells which don't satisfy the condition are removed and transformed to unuseful cells. The final clustering result only contains useful cells.

Definition 4 (Neighbor cells) In 2 dimensional data space, we assume that cell $C_{i_1 i_2}$ and $C_{j_1 j_2}$ are neighbor cells, m is the number of intervals in each dimension. The neighbor cells satisfy inequation 1. Figure 1 shows the relationship between a current cell and its neighbor cells corresponding to this inequation.

$$\left| i_p - j_p \right| \le 1 , (p = 1,2; 1 \le i, j \le m) \quad (1)$$

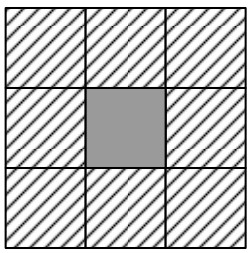

Figure 1. The current gray cell and all its neighbor cells

3. Density Confidence Interval of a Useful Cell

The characteristic of confidence interval [7] is that for a given dataset the interval reflects the general tendency of these data. If one data is abnormal from most of the others, it will not be included into such interval. Similarly, each useful cell has a density confidence interval about its neighbor cells and itself by calculating the densities of these cells. If the density of the current cell is smaller than the lower limit of its density confidence interval, the current cell should not be included into the local dense area. It should be treated as an unuseful cell. By contraries, if its density is larger than the upper limit of its density confidence interval, it is kept as a useful cell, which is one part of a local dense area. Compared with setting one density threshold on the whole dataset, this method has the ability to recognize local dense areas in terms of each useful cell's density thresholds in the quantized space where the multi-density clusters exist.

The density confidence interval of a useful cell is calculated by using equation 2, 3 and 4. Firstly, equation 2 is used to calculate the mean density value of the current cell and its useful neighbors. Secondly, the variance can be obtained by using equation 3. At last, the density confidence interval is calculated as what equation 4 shows.

$$\overline{D} = \frac{\sum_{i=1}^{n} D_i}{n} \quad (2)$$

$$Var(\overline{D}) = \frac{\sum_{i=1}^{n} (D_i - \overline{D})^2}{n(n-1)} \quad (3)$$

$$[\overline{D} - t_{n-1,1-\alpha/2}\sqrt{Var(\overline{D})}, \overline{D} + t_{n-1,1-\alpha/2}\sqrt{Var(\overline{D})}] \quad (4)$$

In the above equations, D_i means the density of a useful cell. n is the number of the current cell and its useful neighbor cells. α is a confidence factor and we set it as a fixed value 0.1, because we want to obtain an interval with the probability of 95% to contain the useful cells. $t_{\upsilon,\gamma}$ has the t-distribution with υ degree of freedom. In GDCIC, γ is equal to 1-α/2, namely 0.95 and υ is equal to n-1. Some regular values of t are listed in Table 1 [6].

Table 1. t values for different v and y

γ \ υ	8	7	6	5	4	3	2
0.95	1.860	1.895	1.943	2.015	2.132	2.353	2.920

4. GDCIC Algorithm

According to the above definition, GDCID algorithm is described as follows:

Algorithm: Grid-based Density-Confidence-Interval Clustering algorithm for 2-dimensional multi-density dataset.

Input: the number of data points

Output: clusters

Step 1. construct grid cell according to the number of data points.

Step 2. read the dataset into memory and assign the data to the corresponding cell.

Step 3. calculate the density of each cell and get all useful cells.

for (each cell) {
calculate the density of current cell.
if current cell contains at least one point, tag the cell as useful.
}

Step 4. calculate the density-confidence-interval of each useful cell and remove unreasonable cells.

4.1 calculate the density confidence interval of each useful cell.

4.2 for (each useful cell) {
if the density of the current cells < the lower limit of its confidence interval, tag the cell as unuseful.
}

Setp 5. combine useful neighbor cells to form clusters. If the result doesn't satisfy the requirements, go to step 4.

In this algorithm, setting the number of intervals is a very critical procedure. If the size of each cell is too large, the algorithm will merge two or more clusters into one. Another drawback is that even though it can find the cluster at the right place, it still has so many blank spaces in the large cell. So the result is not exact and satisfying. If the size of each cell is too small, the algorithm may cause the cell number equal or close to the number of data points. For large datasets, the cost of calculation is too expensive even though we regard each cell as the minimum unit of clustering. So it's necessary to find a method to set a suitable interval value to get both a better clustering result and a good efficiency. In 2-dimension data space, we divided each dimension into equal number of intervals. Here, we presume that each dimension will be divided into m intervals, so m^2 grid cells are formed. For each data point in the data set, it will be assigned into the corresponding grid cell according to its coordinate in each dimension. Here we adopt the method which is used in GDILC [8]. Equation 5 is calculated to obtain the number of intervals m in 2 dimensional data space.

$$m = \sqrt{\frac{n}{coefM}} \qquad (5)$$

In equation 5, n is the number of point spatial data. coefM is a positive integral coefficient to adjust the value of m. In fact, it stands for the average number of data points in a cell. But its value is not fixed. In the experiments, we find the ratio between the ideal interval number and the number of data points is not linear. So coefM should be adjusted to obtain a reasonable value of m according to the number of data points. Figure 2 presents the coincidence relationship between coefM and n which is used in our experiments. A good clustering result can be obtained if the value of coefM changes according to such curve.

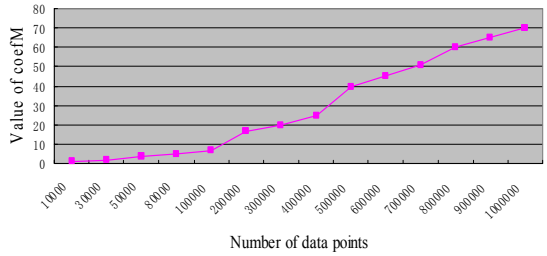

Figure 2. coefM depends on the number of data points n

5. Time Complexity

2-dimension data set is used to analyze the time complexity. As what mentioned in previous part, the data set is scanned for once time to assign each data point into the corresponding cell. Because of the sub-procedures of computing the confidence interval and removing outliers, all the grid cells have to be scanned for two times. So the time complexity of GDCIC is $O(N+2m^2)$. Here, N is the number of data points and m is the number of intervals in each dimension. Because the value of m2 is smaller than the value of N along with the increasing of N, the time complexity of GDCIC is smaller than $O(3N)$. Namely, the time complexity can be treated as $O(N)$. The experiments in Figure 3 also show that the time cost of GDCIC is not linearly increasing along with the increasing number of

data points. Compared with the time complexity of Chameleon and SNN which the value is $O(N^2)$ [5], GDCIC is efficient especially for large data set.

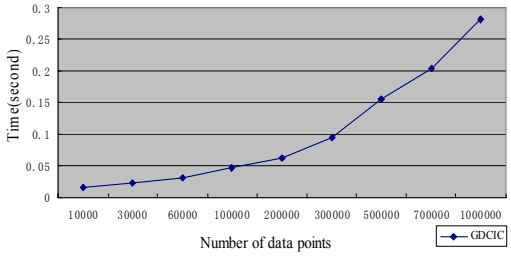

Figure 3. Time cost of GDCIC for different numbers of data points

6. Experimental Results

Experiments are performed by using a personal computer with 512Mb of RAM, Intel Pentium(R) 4 CPU 2.40GHz and running Windows XP Professional. In the synthetic data set, the data points are generated randomly according to certain kinds of distributions. Outliers are randomly distributed. And the clustering results are tagged by different colors. Because SNN shows higher performance than Chameleon [5], we only show the comparison between GDCIC and SNN.

Figure 4(a) is the clustering result of SNN on a multi-dataset, while Figure 4(b) is the clustering result of GDCIC on the same dataset. Compared with SNN, GDCIC can not only recognize the correct clusters, but also eliminate outliers effectively.

Figure 5(a) shows a data set with 120,000 data points, including 20,000 outliers. Figure 5(b) is the clustering result which shows that GDCID can distinguish clusters with arbitrary shapes.

Let two clusters depicted in Figure 6(a). The rightmost consists of 50,000 points and the leftmost of 1000. The density of the small cluster is about two times higher than that of the larger cluster [3]. Figure 6(b) shows the clustering result of GDCIC. Compared with the algorithm of Indexed-based density biased sampling, GDCIC can present an accurate clustering result, but the former algorithm can only get some samples for each cluster.

From the above results, we can know that GDCIC can recognize correct clusters especially for multi-density clustering and eliminate outliers effectively. And the time complexity of GDCIC is $O(N)$.

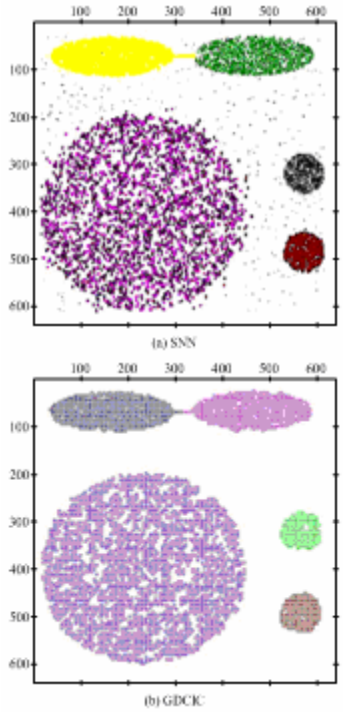

Figure 4. The comparison of clustering result of SNN and GDCIC

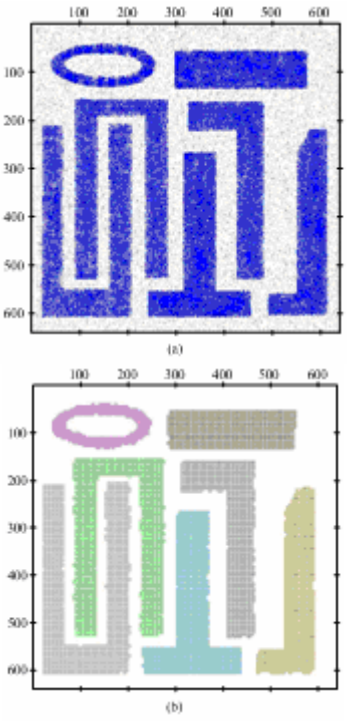

Figure 5. Synthetic dataset and clustering result

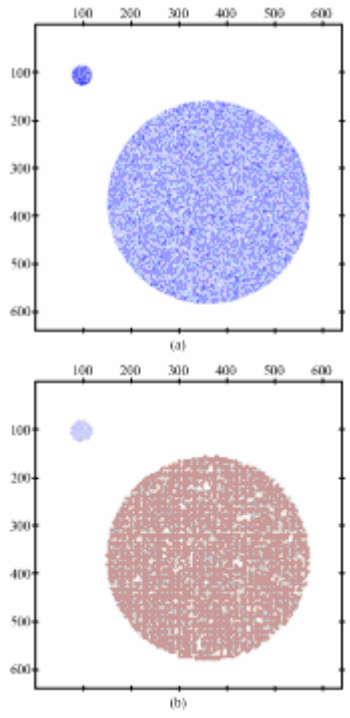

Figure 6. An example of two clusters of different Densities

7. Conclusion

In this paper, a grid-based density-confidence-interval clustering algorithm GDCIC is proposed. For each useful cell, the lower limit of the density confidence interval is regarded as a local density threshold. This algorithm can recognize the local dense area efficiently, and avoids losing the clusters in the data space to a certain degree. Moreover, it is not affected by the outliers and it is order-insensitive. The experiment results show that, GDCIC is preferable to be used for multi-density dataset.

8. Acknowledgement

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment), the Program for New Century Excellent Talents in University (NCET).

9. References

[1] Shashi Shekhar, Sanjay Chawla, <<Spatial Databases: A Tour>>, 2003 by Pearson Education, Inc.

[2] Jiawei Han, Micheline Kamber, <<Data Minging: Concepts and Techniques>>, 2001 by Academic Press.

[3] Alexandros Nanopoulos, Yannis Theodoridis, Yannis Manolopoulos, "Indexed-based density biased sampling for clustering applications," Data & Knowledge Engineering, Volume 57, Issue 1, April 2006, Pages 37-63.

[4] Karypis, G., Eui-Hong Han, Kumar, V., "Chameleon: A Hierarchical Clustering Algorithm Using Dynamic Modeling," IEEE Computer, Volume 32, Issue 8, Aug. 1999 Page(s): 68 – 75.

[5] Levent Ertoz, Michael Steinbach, Vipin Kumar, "Finding Clusters of Different Sizes, Shapes, and Densities in Noisy, High Dimensional Data," SIAM International Conference on Data Mining (SDM '03).

[6] Ho Seok Kim, Song Gao, Ying Xia, Gyoung Bae Kim, Hae Young Bae, "DGCL: An Efficient Density and Grid Baed Clustering Algorithm for Large Spatial Database", Advances in Web-Age Information Management (WAIM 2006) , pp:362-371.

[7] Charles J. Stone, <<A Course in Probability and Statistics>>, ISBN: 0-534-23328-7.

[8] Zhao Yanchang, Song Junde, "GDILC: a grid-based density-isoline clustering algorithm," Volume 3, 29 Oct.-1 Nov. 2001 Page(s):140 - 145 vol.3 Digital Object Identifier 10.1109/ICII.2001.983048.