Efficient Grid-based Clustering Algorithm with Leaping Search and Merge Neighbors Method

Feng Liu[1], Peng Wen[1] and Erzhou Zhu[1, a]

[1]School of Computer Science and Technology, Anhui University, Hefei 230601, China.

E-mail:[a] ezzhu@ahu.edu.cn

Abstract. The increasing data size makes the research of clustering algorithm still an important topic in data mining. As one of the fastest algorithms, the grid clustering algorithm now still suffers from low precision problem. And the efficiency of the algorithm also needed improvement. In order to cope with these problems, this paper proposes an efficient grid-based clustering algorithm by using leaping search and Merge Neighborhood (LSMN). In the algorithm, the LSMN first divides the data space into a finite number of grids and determines the validity of the grid according to the threshold. Then, leaping search mechanism is used to find valid grids of the grid by retrieving all the odd columns and odd rows. Finally, if the number of valid grids is greater than the invalid grid, the invalid grids are merged together. In the algorithm, the time cost is reduced and the accuracy is improved by leaping search and re-judgment of the invalid grid mechanisms respectively. The experimental results have shown that the proposed algorithm exhibits relatively better performance when compared with some popularly used algorithms.

1. Introduction

With the advent of massive data era, it is urgently needed to accurately excavate usefulness data effectively. As a very important and useful technology in data mining, clustering technology can improve the efficiency of data mining. So, it still attracted many focuses from research and industry communities. Up to now, many clustering algorithms have been proposed. Generally, the categories of clustering algorithms can be divided into partition clustering algorithm, hierarchical clustering algorithm, density-based clustering algorithm, grid-based clustering algorithm and hybrid algorithm. Among all kinds of the clustering algorithms, the density-based clustering algorithm and the grid-based clustering algorithms are the two most important ones.

The density-based clustering algorithm is insensitive to noise points which can effectively remove noise points and process any shape of the cluster. DBSCAN [1] is a classic density-based clustering algorithm. DBSCAN is highly accurate, but the operating efficiency is low.

Generally, grid-based clustering algorithms are the most computationally efficient ones. The grid clustering algorithm first divides the data space into a finite number of grids. Then, the grids are divided into valid grids and invalid grids according to the input threshold parameters. Finally, all valid grids in the data space are searched in order and all neighborhood valid grids are merged into clusters. Since the grid clustering algorithm transforms the data points in processing into the grid, so the time complexity of the algorithm depends on the number of grids. When the number of grids is much smaller than data points, the time complexity of the algorithm will be significantly reduced. The classical grid-based clustering algorithms are STING [2] and CLIQUE [3].

Although the grid-based clustering algorithm is an important clustering algorithm, but the way that searching all the grids in the data space in order will seriously increase its execution time. Moreover, there could be some errors when the validity of the grid is only determined by the threshold. In order to improve the speed of grid searching and the quality of clustering, an efficient grid-based clustering algorithm, called LSMN, is proposed in this paper. By LSMN, the odd columns and odd rows in the grid are leaping search. Then the searched valid grids are taken as the starting points to merge the neighborhood valid grids. During this process, when the number of valid grids is greater than the invalid ones, invalid grids are merged together.

2. Lsmn clustering algorithm

Grid-based clustering algorithms always search for rows on a search grid, which undoubtedly reduces the efficiency of the algorithm. Aiming at remarkably reducing the grid searching time cost, this paper proposes a grid-based and density-based clustering algorithm by leaping searching and merging neighborhood (LSMN) grids. After dividing the data space into a grid according to the input parameter ε (grid step size), the input parameter Minpts (threshold) is used to determine whether the grid is valid or invalid. Then, the odd columns and odd rows among all the grids are sequentially searched. During this process, if an invalid grid is found, it is marked as a searched grid. When searching for a valid grid, its neighbor grids are put into a group, and 4 directions leaping grid searching is performed Depending on whether the grid is the valid one or been flagged, iterative searched grids are merged. In order to prevent the situation that leaping searched grids are all invalid grids, the LSMN provides an offset mechanism. By this mechanism, the directly neighbor valid grids become the center for continue grid searching. When all the odd columns and odd rows are searched, the data is clustered. The LSMN algorithm not only can reduce the grid search time but also reduce the chance that the grid is mistaken for an invalid grid.

2.1 Load Data Set and Input Parameters

At the beginning of the algorithm, the ε and Minpts parameters needed to be set. (1) Grid step size (ε): the purpose of the ε parameters is to divide the data set into non-overlapping grids. The appropriate ε parameters can improve the quality of the cluster. (2) The threshold (Minpts): the validity of the grid is determined by Minpts parameters. If the number of data points in the grid exceeds the parameter Minpts, then the grid is defined as a valid grid. Other grids are defined as invalid grid, and the points in the invalid grid are defined as noise.

2.2 Divide the Grid Space

The grid space is divided into non-overlapping grids by parameter ε, and the number of grids is the value of ε multiplies ε. Fig.1 shows all data points classified in the grids.

2.3 Filter Out Noise Points

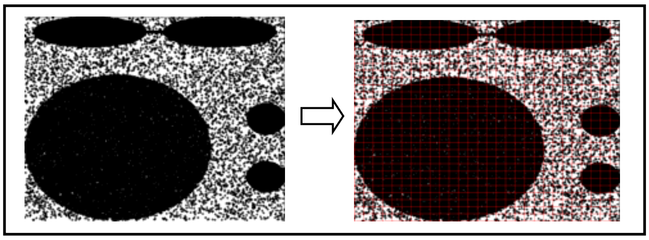

FIGURE 1. Dividing the grid space.

Fig.2 shows the original data set. The "valid" and "invalid" grids in the data space are judged according to the parameter Minpts and the point in the invalid grids is temporarily marked as noise points.

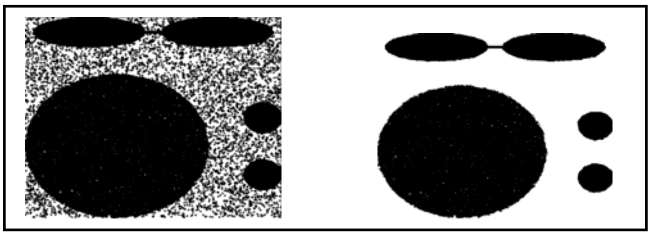

FIGURE 2. The original data sets and the data sets without noise data points.

2.4 Sequentially Search All Odd Columns and All Odd Rows of Grids

The searched grid may be valid grid or invalid grid. When the searched grid is invalid one, mark it and search for the next grid. When the searched grid is valid grid and not been marked, the neighboring all valid grids are merged with it. If the invalid grid is less than three, merge them together into the cluster. Then, leaping searching grids along four directions (as shown in Fig.3).

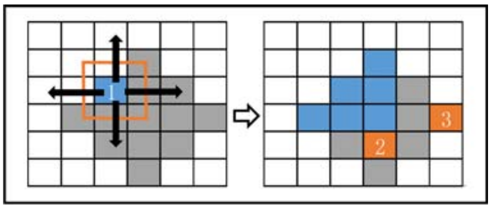

FIGURE 3. Valid grid processing.

During this process, if a valid and not marked grid is encountered, continue merging neighbor grids and leaping searches (as shown in Fig.4)

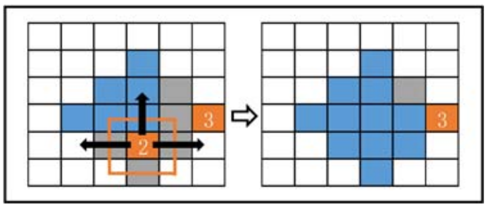

FIGURE 4. Leaping search for valid grid.

In order to prevent the situation appeared in Fig.5, the algorithm provides an offset mechanism. If four grids that leaping searched are all invalid, the directly neighbor valid grids become the center to continue to search.

2.5 Complete Clustering of Data Points

After all odd columns and all odd rows of grids are searched, clustering of data points is completed.

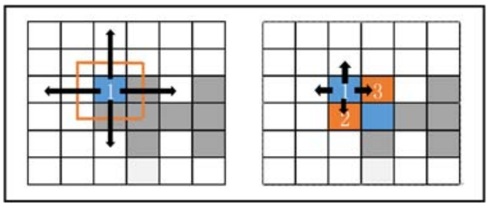

FIGURE 5. Offset operation.

3. Performance of lsmn

All experiments in this paper are carried out in JAVA program, and run on Microsoft Windows 7 desktop computer with Intel i3 CPU (3.7GHz) equipped with 4GB RAM.

The efficiency and quality of LSMN are verified by comparing the experimental results with 3 popularly used clustering algorithms (DBSCAN, CLIQUE and RTHVDS [4]).

Table 1 lists the details of the 4 teat data sets. Table 2 shows the parameter values.

Table 3 shows the experimental results. All algorithms for each dataset are tested 10 times, and average time, accuracy were calculated. DBSCAN has the highest accuracy, but as the number of data sets increases, the time consumption increases dramatically. Although LSMN is less accurate than DBSCAN, it is significantly reduced in time consumption. Compared to CLIQUE and RTHVDS, LSMN is also improved in time and precision. Fig.6 shows the original data sets for the four groups and depicts the clustering results by our LSMN algorithm using four groups of data set.

In addition, LSMN's clustering time is reduced by more than 10% compared to CLIQUE and HVDTHS. And the more the grid number, the faster the clustering time. The increase in algorithm time in the Table 3 is not obvious because the time it takes to read the data is too much. In fact, the efficiency of LSMN has been greatly improved.

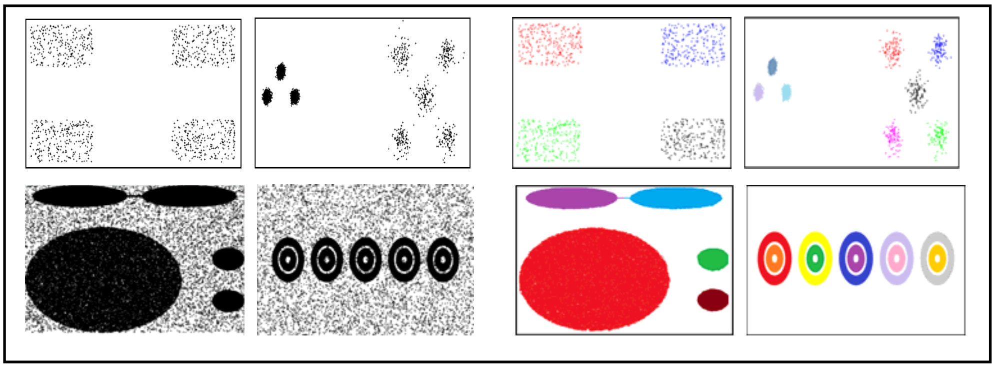

FIGURE 6. The original data set and the clustering results by LSMN.

4. Conclusion

In order to improve the efficiency and quality for clustering mass data, this paper proposed a clustering algorithm based on grid and density. By the algorithm, time cost was reduced by leaping searching; the accuracy is improved by neighbor grids merging. Compared with the DBSCAN algorithm based on density, LSMN greatly improves the running efficiency. Compared with the CLIQUE algorithm and RTHVDS algorithm, the clustering efficiency and precision are improved.

TABLE 1. Information on Four Groups of Data Set

Information	DataSet-1	DataSet-2	DataSet-3 [5]	DataSet-4 [6]
Data Source	By ourselves	M. Rezaei and P. Frnti,2016	TSAI and CHIANG,2016	TSAI and ZHANG,2012
Total number of data point	1000	6500	115000	575000
Number of noise point	0	0	15000	75000
Number of clusters	4	8	5	10

TABLE 2. Parameter Values

Parameter	DataSet-1	DataSet-2	DataSet-3	DataSet-4
ε	20	60	60	200
Minpts	2	2	20	20

TABLE 3. Experimental Results (Run-Time in Second)

Algorithm	Item	DataSet-1	DataSet-2	DataSet-3	DataSet-4
DBSCAN	Run-time	0.29	60	483	27502
	Accuracy	100.00%	100.00%	98.46%	99.46%
CLIQUE	Run-time	0.042	0.119	0.264	0.551
	Accuracy	98.50%	99.70%	98.81%	99.71%
RTHVDS	Run-time	0.039	0.104	0.256	0.545
	Accuracy	99.50%	99.78%	98.87%	99.54%
LSMN	Run-time	0.028	0.098	0.251	0.539
	Accuracy	100.00%	99.75%	98.91%	99.63%

References

[1] M. Ester, H.P. Kriegel, X. Xu, “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”. International Conference on Knowledge Discovery & Data Mining, 1996, pp.226-231.

[2] W. Wang, J. Yang, R.R. Muntz, “STING: A statistical information grid approach to spatial data mining”. International Conference on Very Large Data Bases, 1997, pp.186-195.

[3] R. Agrawal, J. Gehrke, D. Gunopulos, et al, “Automatic subspace clustering of high dimensional data for data mining application”. Data Mining & Knowledge Discovery, 1998, pp.94-105.

[4] J.D. Zhao, Y.W. Yu, J.L. Liu, “A Data Clustering Algorithm over Real Time High-Volume Data Streams”. Journal of Beijing University of Posts and Telecommunications, 2016, pp.114-119.

[5] C.F Tsai, C. Yao, “ENHANCEMENT OF DATA CLUSTERING USING TSS-DBSCAN APPROACH FOR DATA MINING”. International Conference on Machine Learning & Cybernetics, 2016, pp.535-540.

[6] C.F Tsai, J.H. Zhang, “Grid Clustering Algorithm with Simple Leaping Search Technique”. International Symposium on Computer, 2012, pp.938-941.