GF-DBSCAN: A New Efficient and Effective Data Clustering Technique for Large Databases

CHENG-FA TSAI, CHIEN-TSUNG WU
Department of Management Information Systems, National Pingtung University of Science and Technology, Pingtung, TAIWAN
E-mail: cftsai@mail.npust.edu.tw

Abstract: The DBSCAN data clustering accurately searches adjacent area with similar density of data, and effectively filters noise, making it very valuable in data mining. However, DBSCAN needs to compare all data in each object, making it very time-consuming. This work presents a new clustering method called GF-DBSCAN, which is based on a well-known existing approach named FDBSCAN. The new algorithm is grid-based to reduce the number of searches, and redefines the cluster cohesion merging, giving it high data clustering efficiency. This implementation has very high clustering accuracy and filtering rates, and is faster than the famous DBSCAN, IDBSCAN and FDBSCAN schemes.

Key-word: data mining, data clustering, database, density-based clustering, grid-based clustering, algorithm

1 Introduction
Data storage capacity is growing owing to advances in information technology. Hence, obtaining information quickly and correctly is a biggest challenge. Data mining technology has thus become important. Data clustering is a commonly used way of finding the different groups of similar data. Data in the same group are highly similar, and data in different groups are dissimilar. Cluster analysis approaches are categorized as partitioning, density-based, hierarchical and grid-based methods.

One well-known partitioning scheme is K-means [1], in which a cluster is assigned parameter, and then regarded the near data as the same cluster, K-means is very fast and easy to implement, but cannot easily recognize arbitrary shapes.

Density-based clustering algorithm can recognize arbitrary shapes. DBSCAN is a well-known algorithm that searches the neighboring objects of a region, and determines which objects belong to the same cluster [2]. However, DBSCAN requires a long search time. Therefore, IDBSCAN employs the sampling in the search scope to decrease the time cost [3]. FDBSCAN adopts quick merging to reduce redundant searching and raise efficiency [4].

Hierarchical clustering methods compare clusters with each other. These algorithms take a long time in identifying cohesion and division in groups. Well-known hierarchical clustering algorithms include BIRCH [5] and CURE [6].

The grid-based approach is very efficient. Objects are distributed into multiple dimensions. Each grid-cell has many points. Grid-cells are placed into clusters and according to dimension
objects. STING [7] is a famous grid clustering method.

This paper develops a new clustering algorithm, named GF-DBSCAN, which improves FDBSCAN.

2 Related Works
This section illustrates several data clustering algorithms.

2.1 DBSCAN Algorithm
DBSCAN is the first density-based clustering algorithm, and was presented in 1996 by M. Ester et al. The algorithm has two parameters, namely Eps and MinPts, where Eps denotes the region in the object for searching neighborhoods, and Minpts represents the minimum number of objects. In Fig.1, if the neighborhoods of point C have at least MinPts points, then C is called a core point. Points in the search region are clustered repeatedly until fewer than MinPts remain. For instance, Point B is searched from point P, involving fewer than MinPts points. Consequently, point B must be treated as the border point. Point O has fewer than MinPts neighborhoods points, so cannot become a new cluster.

2.2 IDBSCAN Algorithm
The IDBSCAN algorithm was designed to eliminate the sampling on DBSCAN, and was proposed in 2004 by Borah et al. Representative points on the circle form the Marked Boundary Objects (MBO). The MBO identify the eight distinct objects in the neighborhood of an object. The points that are the closest to these eight positions are selected as seeds. Therefore, at most eight objects are labeled at once. IDBSCAN requires fewer seeds than DBSCAN, as indicated in Fig. 2.

2.3 FDBSCAN Algorithm
The FDBSCAN algorithm, which was developed in 2006 by Bing Liu, extends DBSCAN by adding non-linear searching, and has the same parameters as DBSCAN. DBSCAN scans many objects repeatedly with many times. FDBSCAN solves this problem by applying an external scan to reduce the search times. In Fig. 3, Point O is involved in two different clusters P and cluster Q. Additionally, is defined as a core Point. Hence, these clusters P and Q are the same.

FDBSCAN defines clusters by density and reduces the point search action and time cost depending on the point at which clusters are merged.
3 Proposed GF-DBSCAN Scheme

This section introduces the proposed new GF-DBSCAN clustering algorithm, and describes the execution steps in terms of simple images.

3.1 Concept of Improved GF-DBSCAN

The main concept of improved GF-DBSCAN clustering algorithm can be described as follows:

In the DBSCAN and relative density-based clustering approaches, each object must be compared with numerous other objects in the dataset. In order to reduce execution time cost, thus the proposed GF-DBSCAN algorithm adopts grid-based scheme. Each point is assigned to a particular grid-cell, as revealed in Fig. 4.

![Fig. 4. Space partitioning using grid.](image)

Each point search near objects is limited to the neighboring grid-cells. Each data point must be compared with all other data points, leading to a very long execution time. The proposed grid-based scheme reduces the time taken to search points in the complete dataset. The central cells determine the area of neighboring cells, as shown in Fig. 5.

If the points in a new cluster do not involve any other cluster points, then the new cluster is independent of any other cluster. For example, Cluster 1 satisfies at least MinPts points, and thus becomes a new cluster, as demonstrated in Fig. 6(a). Cluster 2 and cluster 3 are new clusters that do not intersect with any other clusters. Therefore, three independent clusters are defined. In fig. 6(b) and 6(c), a new Cluster 4 is defined, which intersects Cluster 1. If the overlapping objects include the core object, then Clusters 1 and 4 should be merged.

Likewise, if the new cluster intersects with many clusters, then these clusters are merged into the previous cluster. In Figs 6(d)-6(f), Cluster 5 is the new cluster, and intersects Cluster 1 and Cluster 2, thus merging clusters to form a single cluster. In sum, the proposed GF-DBSCAN algorithm can lower the time cost of searching, and improves efficiency. Fig. 7(a) depicts a case of segmenting data into grid-cells, in which the five clusters are satisfying the MinPts are compared with other data in neighboring cells to generate a new cluster. Those clusters have different areas. The search points of Cluster 3 are seen only in the neighboring cells. In Fig. 7(b), the new cluster embraces the edges of other clusters, which have core points in other clusters. Accordingly, these clusters can be merged. Fig. 7(c) displays the result of clustering, in which three clusters are formed.
3.2 The algorithm of GF-DBSCAN

This section describes in detail the steps of the GF-DBSCAN algorithm.

Step 1. Select datasets and initialization parameters. The GF-DBSCAN clustering algorithm requires input two parameters: the radius (Eps), and the minimum number of points (MinPts), as described below: (1) Radius (Eps): this parameter has two functions. First, the radius of the data points can be considered as the core point of the neighborhood points. Second, the radius determines the grid-cell length of each side. In other words, each cell has the radius Eps. (2) Minimum number of points (MinPts): To form a cluster, the point searching the neighborhoods points must be bigger than or be equal to the points. Otherwise, it is viewed as a noise point.

Step 2. Partitioning the data into several cells. Data are partitioned into cells according to the parameter Eps. Each point must be defined in a specific cell. Each cell has a unique serial number.

Step 3. Read data and search the neighborhoods cells. Read the data in point order until all points are defined, then stop the algorithm. The search scope of a point is obtained as soon as the point is read.

Step 4. Determine whether the number of points is less than MinPts. The number of data points is determined from the radius scope. If the number of points in a group is greater than or equal to the setting parameter of MinPts, then these points are regarded as a cluster. Conversely, a set of points is defined as noise if the points number less than MinPts. In this case, return step 3.

Step 5. Decide whether to create a new cluster or
merge sub-clusters. There are two choices in this step. The first choice is creating a new cluster if the current cluster does not include others clusters, as step 6 revealed. Otherwise, merge the sub-cluster into a single cluster, as depicted on step 7.

**Step 6.** Create a new independent cluster. If the points in a new cluster do not involve any other cluster points, then the new cluster is independent of any other cluster.

**Step 7.** Merge the sub-cluster into a single cluster. The new cluster intersects with other clusters. The intersection point is used to search neighborhoods points again in neighbor cells. If the search points are satisfy less point, meaning that the core point is included within two or more clusters, and then these clusters should be merged. Conversely, clusters whose intersection does not invoke the core point are not merged. Similarly, if a new cluster intersects many clusters, then each intersecting point that has the cluster as its core point is identified.

**Step 8.** Stop the condition. If not all points are defined yet, then return to Step 3 until all points are defined, and stop search.

### 4. Experimental Results

The algorithms were run on a desktop computer with 2G RAM, and an Intel 3.4GHz dual-core CPU on Microsoft Windows XP Professional Operational System, and using Java-based programs. This experiment utilized seven 2-D datasets (Fig. 8) with 115,000, 230,000 and 575,000 objects, and all with 15% noise to compare different algorithms. Fig. 9 shows the experimental results with 575,000 objects using GF-DBSCAN. Additionally, Table 1 lists the clustering results from DBSCAN, IDBSCAN, FDBSCAN and the proposed GF-DBSCAN. The computational time of DBSCAN was higher than those of IDBSCAN, FDBSCAN and GF-DBSCAN. FDBSCAN using fast merging had significantly lower execution time cost, but took a long time to search all data in a large dataset. The proposed GF-DBSCAN has the lowest time cost and fairly good clustering correctness rate and noise filtering rate.

### 5. Conclusion

This work presents a new clustering algorithm, named GF-DBSCAN, which combines fast merging method of hierarchical by density region searching and grid-based limit concept. The experimental results involving seven datasets demonstrate that the GF-DBSCAN has about 1/100 of the time cost of FDBSCAN, and also has the same clustering correctness rate (CCR) and noise filtering rate (NFR) as FDBSCAN. Thus, the proposed algorithm produces stable clustering.
Table 1: Comparisons with GF-DBSCAN, DBSCAN, IDBSCAN and FDBSCAN using 575,000 objects data sets with 15% noise; item 1 denotes execute time of cost (in seconds); item 2 represents the clustering correctness rate (%). Finally, item 3 indicates the noise filtering rate (%).

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Item</th>
<th>DataSet-1</th>
<th>DataSet-2</th>
<th>DataSet-3</th>
<th>DataSet-4</th>
<th>DataSet-5</th>
<th>DataSet-6</th>
<th>DataSet-7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>4257</td>
<td>4271</td>
<td>4245</td>
<td>4253</td>
<td>4210</td>
<td>4212</td>
<td>4249</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>98.72%</td>
<td>98.04%</td>
<td>97.12%</td>
<td>98.12%</td>
<td>97.89%</td>
<td>98.10%</td>
<td>96.85%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1475</td>
<td>2209</td>
<td>1963</td>
<td>1960</td>
<td>2078</td>
<td>2208</td>
<td>2195</td>
</tr>
<tr>
<td>IDBSCAN</td>
<td>2</td>
<td>99.99%</td>
<td>99.91%</td>
<td>99.97%</td>
<td>99.97%</td>
<td>99.99%</td>
<td>99.95%</td>
<td>99.99%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>97.27%</td>
<td>98.81%</td>
<td>97.43%</td>
<td>98.26%</td>
<td>97.59%</td>
<td>98.28%</td>
<td>97.47%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>732.547</td>
<td>774.766</td>
<td>745.656</td>
<td>729.328</td>
<td>733.250</td>
<td>827.031</td>
<td>790.047</td>
</tr>
<tr>
<td>FDBSCAN</td>
<td>2</td>
<td>99.70%</td>
<td>99.78%</td>
<td>99.81%</td>
<td>99.70%</td>
<td>99.95%</td>
<td>99.90%</td>
<td>99.79%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>98.81%</td>
<td>98.92%</td>
<td>98.21%</td>
<td>99.08%</td>
<td>98.23%</td>
<td>98.80%</td>
<td>98.86%</td>
</tr>
<tr>
<td>GF-DBSCAN</td>
<td>2</td>
<td>99.70%</td>
<td>99.78%</td>
<td>99.81%</td>
<td>99.70%</td>
<td>99.95%</td>
<td>99.90%</td>
<td>99.79%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>98.81%</td>
<td>98.92%</td>
<td>98.21%</td>
<td>99.08%</td>
<td>98.23%</td>
<td>98.80%</td>
<td>98.86%</td>
</tr>
</tbody>
</table>

References:


