An Enhanced Density and Grid based Spatial Clustering Algorithm for Large Spatial Database

Song Gao°· Ho Seok Kim°· Ying Xia°· Gyoung Bae Kim°· Hae Young Bae°·

ABSTRACT

Spatial clustering, which groups similar objects based on their distance, connectivity, or their relative density in space, is an important component of spatial data mining. Density-based and grid-based clustering are two main clustering approaches. The former is famous for its capability of discovering clusters of various shapes and eliminating noises, while the latter is well known for its high speed. Clustering large data sets has always been a serious challenge for clustering algorithms, because huge data set would make the clustering process extremely costly. In this paper, we propose an enhanced Density-Grid based Clustering Algorithm for Large spatial database by setting a default number of intervals and removing the outliers effectively with the help of a proper measurement to identify areas of high density in the input data space. We use a density threshold DT to recognize dense cells before neighbor dense cells are combined to form clusters. When proposed algorithm is performed on large database, a proper granularity of each dimension in data space and a density threshold for recognizing dense areas can improve the performance of this algorithm. We combine grid-based and density-based methods together to not only increase the efficiency but also find clusters with arbitrary shape. Synthetic datasets are used for experimental evaluation which shows that proposed method has high performance and accuracy in the experiments.

Key Words : Clustering, Spatial Data Mining, Spatial Database, Outlier

1. INTRODUCTION

Spatial data mining is the discovery of interesting char-

리스 텍스트의 주요 내용은 다음과 같습니다.

**An Enhanced Density and Grid based Spatial Clustering Algorithm for Large Spatial Database**

Song Gao°· Ho Seok Kim°· Ying Xia°· Gyoung Bae Kim°· Hae Young Bae°·

**ABSTRACT**

Spatial clustering, which groups similar objects based on their distance, connectivity, or their relative density in space, is an important component of spatial data mining. Density-based and grid-based clustering are two main clustering approaches. The former is famous for its capability of discovering clusters of various shapes and eliminating noises, while the latter is well known for its high speed. Clustering large data sets has always been a serious challenge for clustering algorithms, because huge data set would make the clustering process extremely costly. In this paper, we propose an enhanced Density-Grid based Clustering Algorithm for Large spatial database by setting a default number of intervals and removing the outliers effectively with the help of a proper measurement to identify areas of high density in the input data space. We use a density threshold DT to recognize dense cells before neighbor dense cells are combined to form clusters. When proposed algorithm is performed on large database, a proper granularity of each dimension in data space and a density threshold for recognizing dense areas can improve the performance of this algorithm. We combine grid-based and density-based methods together to not only increase the efficiency but also find clusters with arbitrary shape. Synthetic datasets are used for experimental evaluation which shows that proposed method has high performance and accuracy in the experiments.

**Key Words**: Clustering, Spatial Data Mining, Spatial Database, Outlier

**1. INTRODUCTION**

Spatial data mining is the discovery of interesting char-

論文의 주요 내용은 다음과 같습니다.

**An Enhanced Density and Grid based Spatial Clustering Algorithm for Large Spatial Database**

Song Gao°· Ho Seok Kim°· Ying Xia°· Gyoung Bae Kim°· Hae Young Bae°·

**ABSTRACT**

Spatial clustering, which groups similar objects based on their distance, connectivity, or their relative density in space, is an important component of spatial data mining. Density-based and grid-based clustering are two main clustering approaches. The former is famous for its capability of discovering clusters of various shapes and eliminating noises, while the latter is well known for its high speed. Clustering large data sets has always been a serious challenge for clustering algorithms, because huge data set would make the clustering process extremely costly. In this paper, we propose an enhanced Density-Grid based Clustering Algorithm for Large spatial database by setting a default number of intervals and removing the outliers effectively with the help of a proper measurement to identify areas of high density in the input data space. We use a density threshold DT to recognize dense cells before neighbor dense cells are combined to form clusters. When proposed algorithm is performed on large database, a proper granularity of each dimension in data space and a density threshold for recognizing dense areas can improve the performance of this algorithm. We combine grid-based and density-based methods together to not only increase the efficiency but also find clusters with arbitrary shape. Synthetic datasets are used for experimental evaluation which shows that proposed method has high performance and accuracy in the experiments.

**Key Words**: Clustering, Spatial Data Mining, Spatial Database, Outlier

**1. INTRODUCTION**

Spatial data mining is the discovery of interesting char-

論文의 주요 내용은 다음과 같습니다.

**An Enhanced Density and Grid based Spatial Clustering Algorithm for Large Spatial Database**

Song Gao°· Ho Seok Kim°· Ying Xia°· Gyoung Bae Kim°· Hae Young Bae°·

**ABSTRACT**

Spatial clustering, which groups similar objects based on their distance, connectivity, or their relative density in space, is an important component of spatial data mining. Density-based and grid-based clustering are two main clustering approaches. The former is famous for its capability of discovering clusters of various shapes and eliminating noises, while the latter is well known for its high speed. Clustering large data sets has always been a serious challenge for clustering algorithms, because huge data set would make the clustering process extremely costly. In this paper, we propose an enhanced Density-Grid based Clustering Algorithm for Large spatial database by setting a default number of intervals and removing the outliers effectively with the help of a proper measurement to identify areas of high density in the input data space. We use a density threshold DT to recognize dense cells before neighbor dense cells are combined to form clusters. When proposed algorithm is performed on large database, a proper granularity of each dimension in data space and a density threshold for recognizing dense areas can improve the performance of this algorithm. We combine grid-based and density-based methods together to not only increase the efficiency but also find clusters with arbitrary shape. Synthetic datasets are used for experimental evaluation which shows that proposed method has high performance and accuracy in the experiments.

**Key Words**: Clustering, Spatial Data Mining, Spatial Database, Outlier

**1. INTRODUCTION**

Spatial data mining is the discovery of interesting char-
commodate data semantics.

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 geographic space. From this point, we propose DGCL to process dataset in spatial database. Usually spatial datasets are very large, such as the earth scientific datasets which include location and climate data of the earth during certain period. The disadvantage of shifting grid and GDILC is their low efficiency for large dataset, even though they can get correct clustering results. And that’s what we want to improve in DGCL.

The aim of data clustering algorithms is to group the objects in spatial databases into meaningful subclasses. A good clustering algorithm should have the following characteristics. First, due to the huge amount of spatial data, an important challenge for clustering algorithm is to achieve good time efficiency. Second, a good clustering algorithm should be able to identify clusters irrespective of their shapes or relative position. Third, it should have better ability to handle noise or outliers. Outliers refer to spatial objects which are not contained in any cluster and should be removed during the mining process. Fourth, it should be order insensitive with respect to input data. Last, some applications may require clustering at hierarchical levels of coarseness which we call the multi-resolution property. To obtain results with different levels, the clustering algorithm may need input parameters, so fewer parameters are convenient for users.

None of the well-known clustering algorithms, such as shifting grid[1] and GDILC[2], satisfies the combination of these requirements, especially the efficiency for large dataset. This paper presents an efficient density-grid based clustering algorithm, DGCL, which can handle huge amount of spatial data with noise efficiently and find natural clusters correctly. When DGCL is performed on large dataset, a proper granularity of each dimension in data pace and a density threshold for recognizing dense areas can improve the performance of this algorithm.

The rest of this paper is organized as follows. Section 2 reviews related work. The detail algorithm is described in Section 3. Section 4 we analyze the time complexity. Section 5 shows the results of experiments. A conclusion is presented in the last section.

2. RELATED WORK

Density-based and grid-based clustering are two main clustering approaches. The former is famous for its capability of discovering clusters of various shapes and eliminating noises, while the latter is well known for its high speed. Shifting grid and GDILC are two kinds of clustering methods which are based on density and grid. Both of them have advantages and disadvantages.

Density-based[3] clustering algorithms regard clusters as dense regions of objects in the data space that are separated by regions of low density. A density-based cluster is a set of density-connected objects that is maximal with respect to density-reachability. Every object not contained in any cluster is considered to be noise. Typical example is DBSCAN. It grows regions with sufficiently high density into clusters and discovers clusters of arbitrary shape in spatial databases with noise. It has two parameters($\varepsilon$, MinPts). But the users usually don’t know clearly about the suitable values of these two parameters for some data sets. Other drawback of this technique is the high computational complexity because of examining all the neighborhoods in checking the core condition for each object. Especially when the algorithm runs on very large datasets, this step is very expensive. Its time complexity is $O(n \log n)$, where $n$ is the number of data objects [4]. Because of this point, it’s also insensitive with the order of input data.

Grid-based[3] clustering algorithms use a multi-resolution grid data structure. It quantizes the space into a finite number of cells that form a grid structure on which all of the operations for clustering are performed. The main advantage is its fast processing time, which is typically independent of the number of data objects, yet dependent on only the number of cells in each dimension in the quantized space.

A shifting grid[1] clustering algorithm does not require users inputting parameters. It divides each dimension of the data space into certain interval to form a grid structure in the data space. Based on the concept of sliding window, shifting of the whole grid structure is introduced to obtain a more descriptive density profile. It clusters data in a way of cells rather than in points [5]. But its processing is a recursive procedure. It may start at one cluster for all data points and become one data point per cluster eventually. If the difference between the result of previous iteration and that of current iteration is less than or equal to an acceptable error, AcceptErr, then the current result will be recognized as the final result. Therefore, for large dataset, this algorithm is time-consuming, even though it can enhance the accuracy of the results.

GDILC[2] is a grid-based density-isoline clustering algorithm. The idea of clustering using density-isoline comes from contour figures, density-based and grid-based clustering algorithms. It has two thresholds, namely dis-
distance threshold RT and density threshold DT. When calculating the density of a data sample, only the number of samples that are less than RT away from it are considered. It is the same with those data samples when combination for clusters. From this point, a grid-based method is involved to reduce the number of pairs of data samples to be considered. DT is used as a threshold to remove outliers. When clustering, density-isoline figure obtained from the distribution densities of data sample can be used to discover clusters. Because this algorithm needs to calculate the distance between each data sample and every data sample in its neighbor cells, the cost is also too high, especially for large data set.

In DGCL, we choose a grid-based method to reduce the number of data objects we have to process. We regard a grid cell as the minimum unit in data space, but not a data point. Compared with the number of data points, the number of grid cells is smaller especially in 2-dimensional space. So it can increase efficiency, but to some extent it loses a little accuracy. The characteristic of density-based method is that it uses certain density threshold to distinguish dense areas with sparse areas, then it can find clusters with arbitrary shape. In DGCL, we use a density threshold DT to recognize dense cells before neighbor dense cells are combined to form clusters. We combine grid-based and density-based methods together to not only increase the efficiency but also find clusters with arbitrary shape.

3. THE DENSITY AND GRID BASED ALGORITHM

In DGCL, each data object is treated as a point in data space. We partition the 2 dimensions data space into non-overlapping rectangle units (grid cells), with each dimension being divided into equal number of intervals of equal length. When calculating the density of a cell, neighbor cells of each cell are also considered to strengthen the similarity of each pair of cells. A density threshold is used to distinguish the dense areas with sparse areas and outliers. Because a suitable granularity of each dimension is set, the clustering procedure need only one time without modifying the granularity again. Compared with shifting grid algorithm and GDILC, there is no iteration in DGCL and it doesn’t need to calculate the distance of each pair of data points in data space. The time complexity is smaller than $O(n)$, which $n$ is the number of data points in data space, yet depends on the number of grid cells.

3.1 Procedure of DGCL

There are 7 steps in DGCL. In step 1, get the total number of the data points. According to the total number, construct the grid cell. In step 2, read the data set from the disk. If there is not sufficient memory to contain the whole data set, this algorithm divides them to several parts and read one part at a time to calculate the density of each cells until all parts have been read for one time. In step 3, calculate the density threshold DT by using the equation in GDILC[2]. The aim is to remove the outliers and empty cells as much as possible in step 4, and the fourth step can be regarded as a pre-clustering. The remainder cells will be regarded as useful cells. In the following steps, only the useful cells will be considered. So the number of useful cells is small compared with the number of data. In step 5, from the remainder cells, assign the adjacent cells to the same group which should be regarded as a cluster. But that’s not the final result, because the fifth step can only find groups in the sub-area, and some groups are adjacent with each other. So, it’s necessary to merge the adjacent groups together to become one group in step 6. There are still some outliers exiting in the groups, so in the last step, DGCL removes the outliers again to optimize the result. In the end, each group is a cluster. The sketch of DGCL is shown in (Figure 1).

1 : Construct grid cell according to the number of data points.
2 : Load data set.
3 : Distribute the data points into cells and calculate the density threshold DT.
4 : Remove outliers and empty cells.
5 : Group assignment.
6 : Group merging procedure.
7 : Optimization.

(Figure 1) The sketch of DGCL

3.2 Number of intervals

In this step, setting the number of intervals is a very critical procedure. If the size of the cells 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 size of the cell. So the result is not exact and satisfying. If the size of cell is too small, the algorithm may make the number of cells equal or close to the number of points. For large dataset, the cost of calculation is too expensive even though we regard the cell as the minimum unit for 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. Here we adopt the method of GDILC[2]. Equation 1 is used to calculate the number of
intervals \( m \)

\[
m = \sqrt{\frac{n}{\text{coef}M}} \tag{1}
\]

In equation 1, \( n \) is the number of data points. \( \text{coef}M \) is a coefficient to adjust the value of \( m \), we propose it as an positive integer. In fact, it stands for the average number of data samples in a cell. But we don’t want to regard it as a fixed value. Because in the experiments, we find the relationship between the number of cells and that of data points is not linear. So \( \text{coef}M \) also need to be adjusted to get a better result. In our experiments, \( \text{coef}M \) changes according to the following curve in (Figure 2), and it can get a better result as we expect.

(Figure 2) \( \text{coef}M \) depends on the number of data points.

3.3 Density of the cell

For simply, we regard the density as the number of data points contained in the cell. The original grid-based clustering algorithm just considers about the density of the current cell itself[6]. The disadvantage is that it may decrease the relationship of neighbor cells which have similar data points. The attributes of a spatial object stored in a database may be affected by the attributes of the spatial neighbors of that object[5]. To improve this situation, we calculate the density of the considered cell with considering the data points in its neighbor cells such as what the shifting grid clustering algorithm does[1] as illustrated in (Figure 3). \( L \) means the width of each cell, and we never consider about the empty cells.

The definition of neighbor cells satisfies inequation 2. Assume that cell \( C_{ip} \) and \( C_{jp} \) are neighbor cells. \( m \) is the number of intervals.

\[
|i_p - j_p| \leq 1, \quad (p = 1, 2; 1 \leq i, j \leq m) \tag{2}
\]

3.4 Selection of the density threshold \( DT \)

The dense cells are surrounded by the sparse cells which are regarded as outliers. From the experiments, we find that the density of the dense cells usually decreases gradually from the core of the cluster to the boundary as illustrated in (Figure 4), and the density of sparse cells is obviously smaller than that of the boundary cells. So before clustering the cell set, we define a measurement to remove the sparse cells. The aim is to decrease the cost of calculation and increase the efficiency. The measurement, we call it density threshold (DT)[2], is defined using equation 3.

\[
DT = \frac{\text{mean(Density)}}{\log_2(m)} \times \text{coef}DT \tag{3}
\]

In equation 3, \( \text{mean(Density)} \) means the average density of all of the cells. \( m \) is the number of intervals. \( \text{coef}DT \) is an adjustable coefficient between 0.7 and 1. Lots of experi

(Figure 3) The cell with gray color is the considered cell. The density of the considered cell is the sum of the density of the grey area and the line-shadowed part.

(Figure 4) Varying grid density within a circle cluster.
ments show that when setting it to 0.95, good clustering result can be achieved in most conditions. All the cells of which the density is smaller than DT will be removed from the cell set. The remainder cells will be regarded as useful cells. In the following steps, the algorithm only considers about the useful cell, so compared with the number of data points, they’re really few enough. The remainder cells also contain some unexpected cells. They will be further removed in step 7 (optimization).

3.5 Group assignment

This procedure starts from the first useful cell. If the considered cell Ci and all its useful neighbors, all belonging to set Si, haven’t been assigned a group ID, the algorithm assigns a new group ID for all the cells in the set Si. If some cells in Si have had group IDs, the algorithm finds the ID of the cell with the maximum density in Si and assigns it to the other cells in Si. This procedure will not stop until all useful cells have been checked. (Figure 5) shows the sketch of group assignment.

1: Do:
2: Select a useful cell Ci
3: If (there is no group assigned to the cell Ci and its neighbors) 
4: Assign a new group to the cell and all its neighbors.
5: Else 
6: Finding proper group ID and assign it to the other cells.
7: While(there are still useful cells which haven’t been checked)

(Figure 5) The sketch of group assignment

3.6 Group mergence

The procedure of group assignment only considers the current cell and its neighbors, so it clusters the similar cells into the same group in a sub-area. In the whole data space, the adjacent groups need to be merged together to form one group as we expected. The sketch of group mergence is shown in(Figure 6).

1: Do:
2: Select one group gi
3: If (there are some groups adjacent with gi)
4: Merge them into one group gi’.
5: Select the smaller group ID among them.
6: Assign this ID to the new group gi’.
7: While(there are still groups which haven’t been checked)

(Figure 6) The sketch of group mergence

3.7 Optimization

As we mentioned in the previous part, after step 4, the remainder cells also contain some cells which we have not expected.

In DGCL, we calculate the cell density with considering about its neighbors. There are three kinds of situations of neighbors in 2-D space as illustrated in (Figure 7). The problem is that if the considered cell only contains a few data points, its useful neighbors contribute so much to calculate its density. It also includes into a group. But in fact, this cell should be regarded as a cell which contains outlier, and lots of experiments show that these situations often happen at the boundary of the group. (Figure 8) shows one situation of the problem. The fusceus cell is the considered cell. It only contains a relatively few data points compared with its neighbors when calculating the density. So this cell should be removed. The removed cells are regarded as unuselul cells, and the final result only contains the useful cells. We remove the cells which satisfy in equation 4.

\[
RD = \frac{TD}{Num + 1}
\]  

(Figure 7) Situations of neighbors: (a) 3 neighbors in the corner and (b) 5 neighbors at the edge and (c) 8 neighbors inside

(Figure 8) The fusceus cell should be removed
In equation 4, $RD$ means the real number of data points which are contained in the considered cell. $TD$ means the total density of the considered cell. $Num$ is the number of its useful neighbors and 1 means the considered cell itself.

4. TIME COMPLEXITY

Because equation 1 is used to calculate the number of intervals, the total number of cells is $n^2$, namely $n/coefM$, in which $n$ is the number of data points. After we get the data set, both the data distribution and removing outlier procedure need check or calculate all the cells. After that, only relatively few cells need to be checked for group assignment, merge procedure and optimization. So in most of the situations, the time complexity is smaller than $O(n)$. It much more depends on the number of grid cells.

5. EXPERIMENTAL RESULTS

We performed experiments using a personal computer with 768MB of RAM and Pentium(R) 4 CPU 1.8GHz and running Windows XP Professional. The data sets are generated ourselves. In the data set, the data points are generated randomly according to certain kinds of distributions. Noises are randomly distributed. The clustering results are tagged by different colors. (Figure 9) (a) shows a data set with 100,000 data points, including 30,000 for circle, 40,000 for rectangle, 25,000 for sine curve and 5,000 noises.

The result shows that DGCL can find the clusters correctly and eliminate outliers efficiently. Furthermore it also fast enough. If users want to get more exact result, they can adjust the coefficient $coefM$. The smaller the $coefM$ is, the more exact result DGCL gets. But it need more time for calculation. Because GDILC has higher performance than shifting grid algorithm, we only show the comparison of DGCL and GDILC in (Figure 10). From that

(Figure 9) (a) test data set includes 100,000 data points with 5,000 noises and (b) clustering result (3 clusters)

(Figure 10) Comparison between DGCL and GDILC

(Figure 11) clustering results in different levels. (a) data set, 55,000 data points with 2,500 noises and (b) $coefM = 30$, 1 cluster and (c) $coefM = 15$, 2 clusters and (d) $coefM = 4$, 3 clusters

(Figure 12) Datasets and clustering results (a) 20,000 data points with 1,000 noises and (b) 1 cluster and (c) 250,000 data points with 10,000 noises and (d) 3 clusters
graph, we know that this algorithm has higher performance than GDILC.

(Figure 11) shows the results of one data set in different level by adjusting the coefficient coefM. Other data sets are also used to test this algorithm, the result is shown in (Figure 12).

6. CONCLUSION

In this paper, we present an efficient density and grid based clustering algorithm, DGCL. It applies a grid-based method to calculate the density of each cell and can calculate automatically the granularity of each dimension according to the size of a given data set. In DGCL, a proper density threshold is calculated to distinguish dense cells with sparse cells. Adjacent dense cells are combined to form clusters. Compared with the shifting grid clustering algorithm and GDILC, it's much faster because it doesn't need to cluster recursively or calculate the distance between data points. The time complexity depends on the number of grid cells. Moreover, it is not affected by the outliers and can handle them properly. Drawback of this algorithm is that all the cluster boundaries are either horizontal or vertical. A faster method on I/O, such as an efficient indexing method or parallel control, will make the algorithm a whole lot faster. User can get different scale clustering results by adjusting the coefficient coefM. Experiment on synthetic datasets shows that DGCL is a stable and efficient clustering algorithm for large data set.

In future research, experiments will be performed both real and high dimensional dataset.

REFERENCES

[3] Jiawei Han, Micheline Kamber, <<Data Mining: Concepts and Techniques>>, 2001 by Academic Press

Song Gao

e-mail: gao_fly@hotmail.com
2004 Qingdao University, College of Information and Engineering, Computer Science and Technology, Bachelor
2004-present Chongqing University of Posts and Telecommunications, School of Computer Science and Technology, Sino-Korea Chongqing GIS Research Center, Master Course Student
Interesting Field: Spatial data mining, LBS
김효석  

이메일: hskim@dlab.inha.ac.kr  
2001년 인하대학교 전자공학부  
(공학사)  
2003년 인하대학교 컴퓨터공학부  
(공학사)  
2003년~현재 인하대학교  
컴퓨터 정보공학과(박사과정)  
관심분야: 다양한 데이터베이스, LBS, 유비쿼터스 컴퓨팅,  
RFID 비밀화

김성배  

이메일: ghkim@seowon.ac.kr  
1993년 인하대학교 전자공학부  
(공학사)  
1996년 인하대학교 컴퓨터공학부  
(공학석사)  
2000년 인하대학교 전자공학부  
(공학박사)  
2000년~2004년 한국전자통신연구원 신기술연구원  
2004년~현재 서원대학교 컴퓨터공학과 교수  
관심분야: 이동 실시간 데이터베이스, 스프리치 시스템, GIS,  
VOD

Ying Xia  

이메일: xiaying@cqupt.edu.cn  
1993년 Chongqing University of Posts and  
Telecommunications, Department of  
Computer, Major in Computer and  
Telecommunication, Bachelor.  
2001년 Inha University in Korea,  
Department of Computer Science and Engineering,  
Master  
2005~현재 Chongqing University of Posts and  
Telecommunications, Sino-Korea Chongqing GIS  
Research Center, Associate Professor.  
관심분야: Database System and Application, GIS  
Middleware and Application, Mobile  
Computing, Spatial Database, LBS

배해영  

이메일: hybae@inha.ac.kr  
1985년 Univ. of Houston 객원교수  
1990년~1994년 인하대학교 전자공학부 교수  
1992년~현재 인하대학교 컴퓨터공학부 교수  
1995년~현재 지능형GIS연구센터 설립장  
2000년~현재 한국공학전연대회 학술원 명예교수  
2004년~2006년 인하대학교 정보통신계열학과 원장  
관심분야: 블록 데이터베이스, 공간 데이터베이스, 지리정보  
시스템, 비디오디자이너 데이터베이스 등