A Deflected Grid-based Algorithm for Clustering Analysis

NANCY P. LIN, CHUNG-I CHANG, HAO-EN CHUEH,
HUNG-JEN CHEN, WEI-HUA HAO
Department of Computer Science and Information Engineering
Tamkang University
151 Ying-chuan Road Tamsui, Taipei County
TAIWAN, R.O.C
nancylin@mail.tku.edu.tw , taftdc@mail.tku.edu.tw , 890190134@s90.tku.edu.tw
chenhj@mail.sju.edu.tw, 889190111@s89.tku.edu.tw

Abstract: - The grid-based clustering algorithm, which partitions the data space into a finite number of cells to form a grid structure and then performs all clustering operations on this obtained grid structure, is an efficient clustering algorithm, but its effect is seriously influenced by the size of the cells. To cluster efficiently and simultaneously, to reduce the influences of the size of the cells, a new grid-based clustering algorithm, called DGD, is proposed in this paper. The main idea of DGD algorithm is to deflect the original grid structure in each dimension of the data space after the clusters generated from this original structure have been obtained. The deflected grid structure can be considered a dynamic adjustment of the size of the original cells, and thus, the clusters generated from this deflected grid structure can be used to revise the originally obtained clusters. The experimental results verify that, indeed, the effect of DGD algorithm is less influenced by the size of the cells than other grid-based ones.

Key-Words: - Data Mining, Clustering Algorithm, Grid-based Clustering, Significant Cell, Grid Structure

1 Introduction

Clustering analysis which is to group the data points into clusters is an important task of data mining recently. Unlike classification which analyzes the labeled data, clustering analysis deals with data points without consulting a known label previously. In general, data points are grouped only based on the principle of maximizing the intra-class similarity and minimizing the inter-class similarity, and thus, clusters of data points are formed so that data points within a cluster are highly similar to each other, but are very dissimilar to the data points in other clusters.

Up to now, many clustering algorithms have been proposed [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13], and generally, the called grid-based algorithms are the most computationally efficient ones. The main procedure of the grid-based clustering algorithm is to partition the data space into a finite number of cells to form a grid structure, and next, find out the significant cells whose densities exceed a predefined threshold, and group nearby significant cells into clusters finally. Clearly, the grid-based algorithm performs all clustering operations on the generated grid structure; therefore, its time complexity is only dependant on the number of cells in each dimension of the data space. That is, if the number of the cells in each dimension can be controlled as a small value, then the time complexity of the grid-based algorithm will be low. Some famous algorithms of the grid-based clustering are STING [11], WaveCluster [12], and CLIQUE [13].

As the above mentioned, the grid-based clustering algorithm is an efficient algorithm, but its effect is seriously influenced by the size of the grids (or the value of the predefined threshold).

If the cell is small, then it needs many cells to be connected into one cluster. And there will also be more connection of cells. In the connection of cells, the number of data points in cell is the major factor to connect or disconnect the cells. So, the more cells, the more effects. And in the same data space, there are more cells, there will be smaller size.

To cluster data points efficiently and to reduce the influences of the size of the cells at the same time, a new grid-based clustering algorithm, called DGD, is proposed here.

The main idea of DGD algorithm is to deflect the original grid structure in each dimension of the data space after the clusters generated from the original grid structure have been obtained. The deflected grid structure is then used to find out the new significant cells. Next, the nearby significant cells are grouped as well to form some new clusters. Finally, these new generated clusters are used to revise the originally generated clusters.

The rest of the paper is organized as follows: In section 2, some famous grid-based clustering algorithms will be introduced. In section 3, the proposed clustering algorithm, DGD algorithm, will be presented. In section 4, some experiments and discussions will be displayed. The conclusions will be given in section 5.

2 Grid-based Clustering Algorithm

In this section, two popular grid-based clustering algorithms, STING [11] and CLIQUE [13], will be introduced.

STING (Statistical Information Grid-based algorithm) (Wang et al., 1997) exploits the clustering properties of index structures. It employs a hierarchical grid structure and uses longitude and latitude to divide the data space into rectangular cells. STING selects a layer to begin with at the beginning.

For each cell of this layer, to label the cell as relevant if its confidence interval of probability is higher than the threshold. We go down the hierarchy structure by one level and go back to check those cells is relevant or not until the bottom level. Return those regions that meet the requirement of the query. And finally, to retrieve those data fall into the relevant cells.

CLIQUE (Clustering In QUEst) (Agrawal et al., 1998) is a density and grid-based approach for high dimensional data sets that provides automatic sub-space clustering of high dimensional data. It consists of the following steps: First, to uses a bottom-up algorithm that exploits the monotonicity of the clustering criterion with respect to dimensionality to find dense units in different subspaces. Second, it use a depth-first search algorithm to find all clusters that dense units in the same connected component of the graph are in the same cluster. Finally, it will generate a minimal description of each cluster.

In fact, the effects of these two algorithms are seriously influenced by the size of the predefined grids and the threshold of the significant cells. To reduce the influences of the size of the predefined grids and the threshold of the significant cells, we propose a new grid-based clustering algorithm which is called A Deflected Grid-based (DGD) algorithm in this paper.

3 A Deflected Grid-based Algorithm

After the grid structure is built, the deflected grid-based algorithm (DGD) deflects the cell margins by half a cell width in each dimension and have the new grid structure and then combine the two sets of clusters into the final result. The procedure of DGD is shown in the following steps.

Step 1: Generate a grid structure.
By dividing into k equal parts in each dimension, the n dimensional data space is partitioned into k^n non-overlapping cells to be the grid structure.

Step 2: Identify significant cells.
Next, the density of each cell is calculated to find out the significant cells whose densities exceed a predefined threshold.

Step 3: Generate the set of clusters.
Then the nearby significant cells which are connected to each other are grouped into clusters. The set of the clusters is denoted as S_1.

Step 4: Deflect the grid structure.
The original grid structure is next deflected by distance d in each dimension of the data space.

Step 5: Generate the set of new clusters.
The step 2 and step 3 are used again to generate the set of new clusters by using the deflected grid structure. The set of new clusters generated here is denoted as S_2.

Step 6: Revise original clusters.
The clusters generated from the deflected grid structure are used to revise the originally obtained clusters as the following steps.

Step 6a: Find each overlapped cluster C_{2j} for $C_{1i} \in S_1$, and generate the rule $C_{1i} \rightarrow C_{2j}$, where $C_{1i} \cap C_{2j} \neq \phi$, $C_{2j} \in S_2$. The rule $C_{1i} \rightarrow C_{2j}$ means that cluster C_{1i} overlaps cluster C_{2j}. Similarity, find each overlapped cluster C_{1i} for $C_{2j} \in S_2$, and also generate the rule $C_{2j} \rightarrow C_{1i}$, where $C_{2j} \cap C_{1i} \neq \phi$.

Step 6b: The set of all the rules generated in step 6a is denoted as R_o. Next, each cluster $C_{1i} \in S_1$ is revised by using the cluster revised function $CR()$. The cluster modified function $CR()$ is shown in fig.1.

Step 7: Generate the clustering result.

After all clusters of S_1 have been revised, S_1 is the set of final clusters.

```
for each $C_{1i} \in S_1$
    Let X' := X;
    Repeat
        oldX' := X';
        For each $Y \rightarrow Z$ in $R_0$ Do
            If $Y \subset X'$ then
                $X' := X' \cup Z$;
                If $Z \in S_1$ then
                    $S_1 := S_1 - \{Z\}$;
            Endif
    Until (oldX' = X');
    $C_{1i}$ := X';
End
```

Fig.1 the CR algorithm

3.1 Example

In this place, the two dimensional example, shown in figure 2, with 600 points is easy to be divided into two clusters. The example goes through the algorithm.

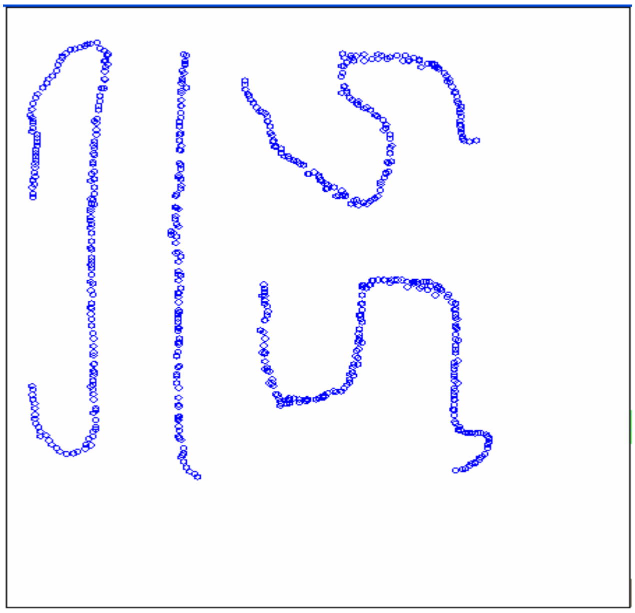


Fig.2 two dimensional example

Step 1: Generate a grid structure.

By dividing into 20 equal parts in each dimension, the two dimensional data space in this example is partitioned into 20^2 non-overlapping cells to be the grid structure, shown in fig.3.

Step 2: Identify significant cells.

Next, the density of each cell is calculated, shown in fig. 4, to find out the significant cells whose densities exceed a predefined threshold, here the threshold is 4.

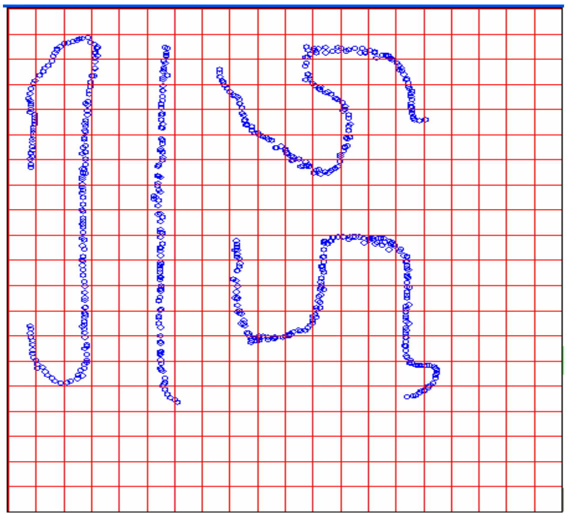


Fig.3 the grid structure of 20^2 cells

	5	8	7		5					5	9	8	10						
2	5	1	6		8		5			8			1	9					
7		6			7		2	8			10	3		9					
8		7			6			10	1			6		6	1				
8		7			7				10	6		9							
2		8			7					8	11								
		6			8														
		7			7														
		6			6			8			12	9	17	4					
		5			8			6			8			9					
		6			9			6			10			9					
3		8			5			10		11	3			8					
7		6			8			6	12					9					
	8	9			7									5	12				
					4	2								5	4				

Fig.4 the density of each cell

step3: Generate the set of clusters.

Then the nearby significant cells which are connected to each other are grouped into 15 clusters. The set of the clusters is denoted as $S_1 = \{C_{11}, C_{12}, \ldots, C_{115}\}$, shown in fig. 5.

step4: Deflect the grid structure.

The original grid structure is deflected by distance d in each dimension of the data space. In this example, d is equal to the half side length of the cell. By deflecting the grid structure, the new one is partitioned into 21^2 cells, shown in fig. 6.

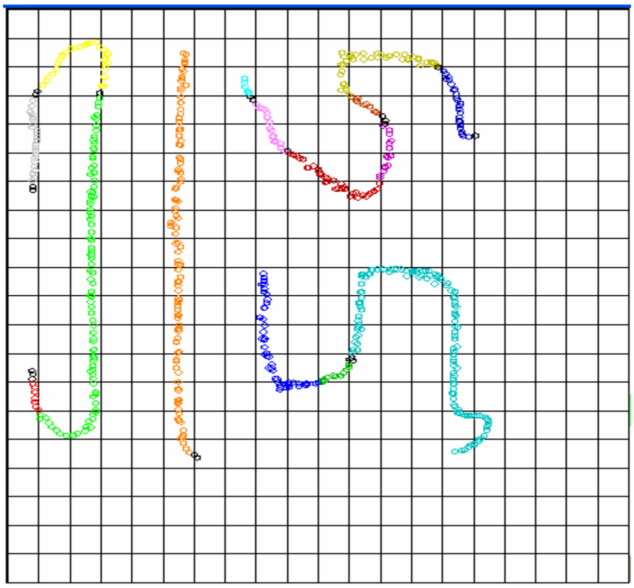


Fig.5 result of first clustering

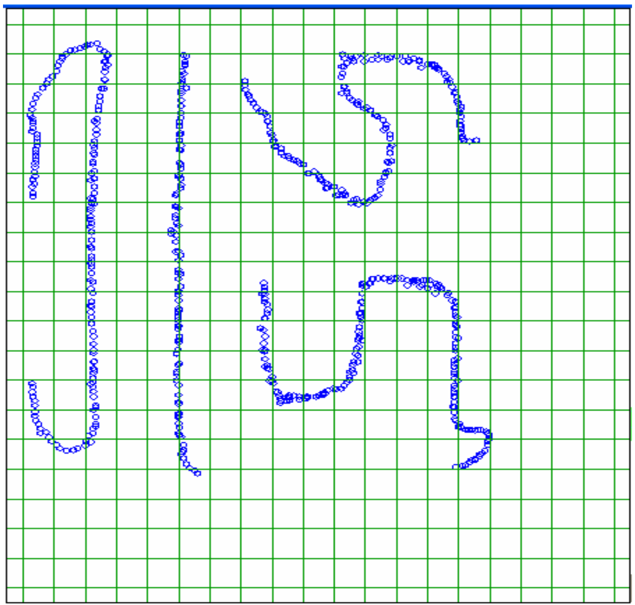


Fig.6 the new grid structure with 21^2 cells

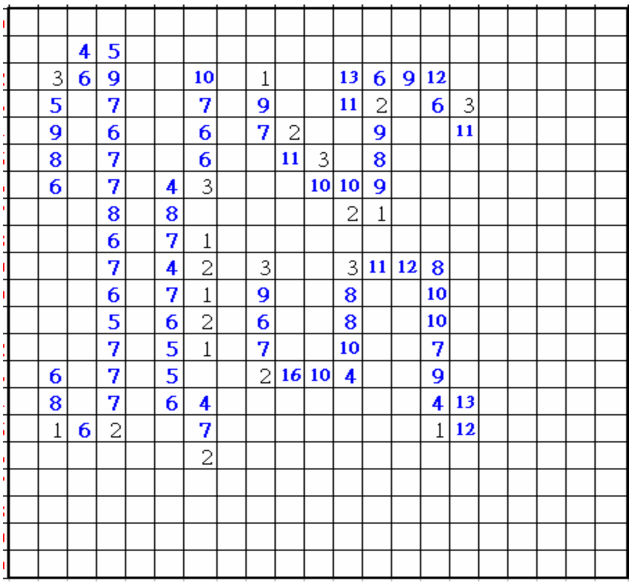
Step 5: Generate the set of new clusters.

Here, the cell density of new grid structure is shown in fig. 7. It's easy to find out the significant cells whose densities exceed a predefined threshold, 4.

And the nearby significant cells which are connected to each other are grouped into 14 clusters. The set of the clusters is denoted as S_2={C21,C22,…,C14}, shown in fig. 8.

Step 6: Revise original clusters.

The clusters generated from the deflected grid structure are used to revise the originally obtained clusters as steps 6.a and 6.b.



		4	5																	
	3	6	9			10		1			13	6	9	12						
	5		7			7		9			11	2		6	3					
	9		6			6		7	2			9			11					
	8		7			6			11	3		8								
	6		7		4	3				10	10	9								
			8		8						2	1								
			6		7	1														
			7		4	2		3			3	11	12	8						
			6		7	1		9			8			10						
			5		6	2		6			8			10						
			7		5	1		7			10			7						
	6		7		5			2	16	10	4			9						
	8		7		6	4								4	13					
	1	6	2			7								1	12					
						2														

Fig.7 the cell density of new grid structure

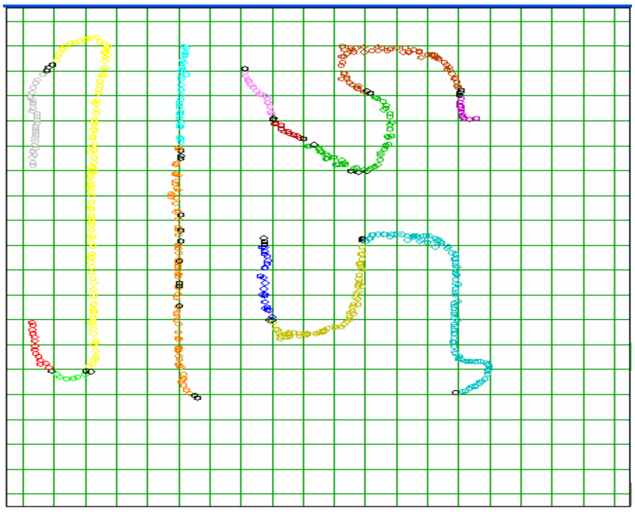


Fig.8 Result of the second clustering

R0 is composed of rules $C_{1i} \rightarrow C_{2j}$, shown in table 1, and $C_{2j} \rightarrow C_{1i}$, shown in table 2.

Step 7: Generate the clustering result.

After all clusters of S_1 have been revised by using cluster modified function $CR()$, revised S_1 is shown in table 3. And the final clustering result is shown in fig. 9.

Manuscript received April 17, 2007; revised March 20, 2008

S_1	Corresponding clusters in S_2	R_0 of S_1
1	1	$C_{11} \rightarrow C_{21}$
2	2	$C_{12} \rightarrow C_{22}$
3	1,3	$C_{13} \rightarrow C_{21}$, $C_{13} \rightarrow C_{23}$
4	3,4	$C_{14} \rightarrow C_{23}$, $C_{14} \rightarrow C_{24}$
5	5,6	$C_{15} \rightarrow C_{25}$, $C_{15} \rightarrow C_{26}$
6	7	$C_{16} \rightarrow C_{27}$
7	7,9	$C_{17} \rightarrow C_{27}$, $C_{17} \rightarrow C_{29}$
8	8,10	$C_{18} \rightarrow C_{28}$, $C_{18} \rightarrow C_{210}$
9	9,11	$C_{19} \rightarrow C_{211}$, $C_{19} \rightarrow C_{211}$
10	12,	$C_{110} \rightarrow C_{212}$
11	10	$C_{111} \rightarrow C_{210}$
12	11,12	$C_{112} \rightarrow C_{211}$, $C_{112} \rightarrow C_{212}$
13	10,13	$C_{113} \rightarrow C_{210}$, $C_{113} \rightarrow C_{213}$
14	11	$C_{114} \rightarrow C_{211}$
15	12,14	$C_{115} \rightarrow C_{212}$, $C_{115} \rightarrow C_{214}$

Table 1 rules $C_{1i} \rightarrow C_{2j}$ of R_0

S_2	Corresponding clusters in S_1	R_0 of S_2
1	1,3	$C_{21} \rightarrow C_{11}$, $C_{21} \rightarrow C_{13}$
2	2,4	$C_{22} \rightarrow C_{12}$, $C_{22} \rightarrow C_{14}$
3	4	$C_{23} \rightarrow C_{14}$
4	4	$C_{24} \rightarrow C_{14}$
5	5	$C_{25} \rightarrow C_{15}$
6	5	$C_{26} \rightarrow C_{15}$
7	6,7	$C_{27} \rightarrow C_{16}$, $C_{27} \rightarrow C_{17}$
8	8	$C_{28} \rightarrow C_{18}$
9	7,9	$C_{29} \rightarrow C_{17}$, $C_{29} \rightarrow C_{19}$
10	8,11,13	$C_{210} \rightarrow C_{18}$, $C_{210} \rightarrow C_{111}$, $C_{210} \rightarrow C_{113}$
11	9,12,14	$C_{211} \rightarrow C_{19}$, $C_{211} \rightarrow C_{112}$, $C_{211} \rightarrow C_{114}$
12	10,12,15	$C_{212} \rightarrow C_{110}$, $C_{212} \rightarrow C_{110}$, $C_{212} \rightarrow C_{115}$
13	13	$C_{213} \rightarrow C_{113}$
14	15	$C_{214} \rightarrow C_{115}$

Table 2 rules $C_{2j} \rightarrow C_{1i}$ of R_0

New C_{1i}	Corresponding original C_{1i} and C_{2j}
C_{11}	C_{11},C_{12},C_{13},C_{14}, C_{21},C_{22},C_{23},C_{24}
C_{12}	
C_{13}	-
C_{14}	-
C_{15}	C_{15},C_{25},C_{26}
C_{16}	C_{16},C_{17},C_{19},C_{110},C_{112},C_{114},C_{115},C_{27},C_{29},C_{211},C_{212},C_{214}
C_{17}	-
C_{18}	C_{18},C_{111},C_{113},C_{28},C_{210},C_{213}
C_{19}	-
C_{110}	-
C_{111}	-
C_{112}	-
C_{113}	-
C_{114}	-
C_{115}	-

Table 3 the set of final clusters

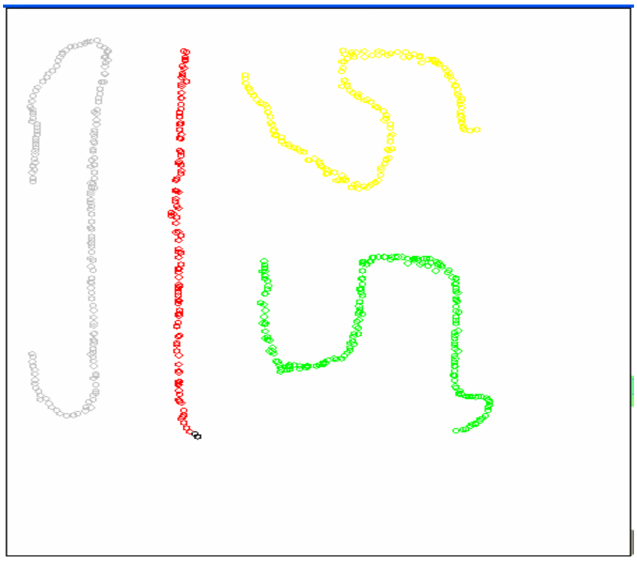


Fig. 9 the final clustering result

4. Experiment and Discussions

Here, we experiment with seven different data. The features are shown in Table 4.

Data	Number of Data	Natural clustering number
Exp 1	600	4
Exp 2	1100	4
Exp 3	1100	5
Exp 4	1150	4
Exp 5	900	3
Exp 6	1000	2
Exp 7	785	3

Table 4 experimental data features

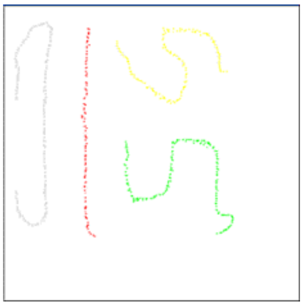


Fig.10 experiment 1

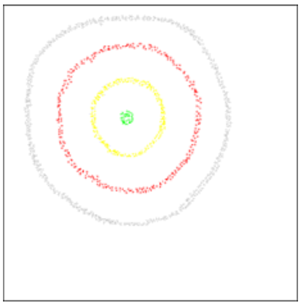


Fig.11 experiment 2

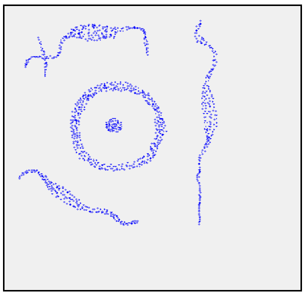


Fig.12 experiment 3

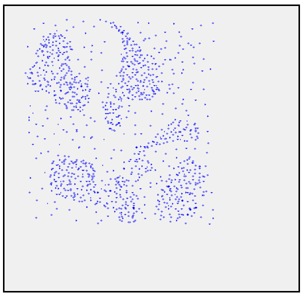


Fig.13 experiment 4

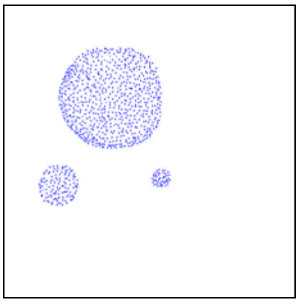


Fig.14 experiment 5

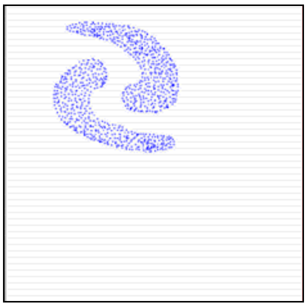


Fig.15 experiment 6

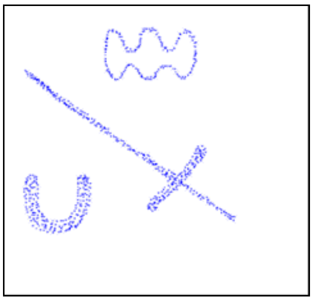


Fig.16 experiment 7

4.1 Experiment

Figure.17 shows the correct rates of DGD and SDG, where the correct clustering result of SDG is by using one of original or new grid structures in the experiment. The correct rates of DGD are all higher than SDG. In the experiments, the correct rates comparison is by using random 100 sets of parameters (density threshold, number of dividing parts in each dimension) from (16, 1) to (55, 3).

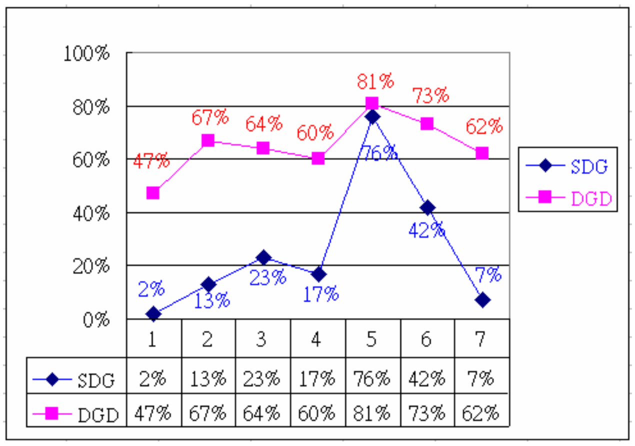


Fig.17 correct rates of DGD and SDG

In table.5, it is the correct rate comparison sheet of experiment 1 by using random 100 sets of parameters. The correct rate of DGD is 47% which is higher than SDG whose correct rate is only 2%. Here, the correct rate of both using the same set of parameters is only 2%. So, the result of SDG is part of the clustering result of DGD in experiment 1. And in experiment 1, it is impossible to find the wrong experimental result that using in DGD but is correct when using in SDG.

SDG DGD	Correct clustering	Incorrect clustering	Rate of DGD
Correct clustering	2%	45%	47%
Incorrect clustering	0%	53%	53%
Rate of SDG	2%	98%	

Table 5 the correct rate comparison sheet of experiment 1

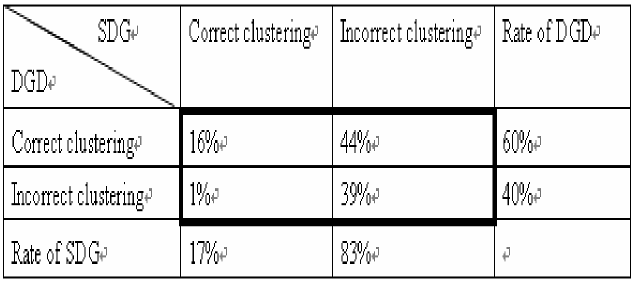
In table.6, 7, 8, and 9, it is possible to find the correct experimental result that using in DGD but is wrong when using in SDG. Though the values are low, the experimental results are not the same as experiment 1 in table 5. So, the results of SDG are not always parts of the clustering results of DGD. Because the correct rate of DGD is always higher than SDG, the experiment by using DGD is able to advance the correct rate than using other grid-based algorithms. In other words, the experimental results verify that the effect of DGD algorithm is less influenced by the size of the cells than other grid-based ones.

SDG DGD	Correct clustering	Incorrect clustering	Rate of DGD
Correct clustering	12%	55%	67%
Incorrect clustering	1%	32%	33%
Rate of SDG	13%	87%	

Table 6: the correct rate comparison sheet of experiment 2

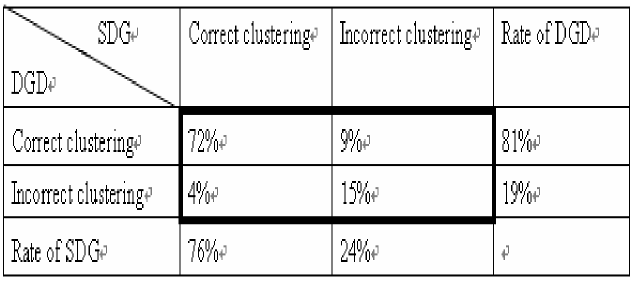
SDG DGD	Correct clustering	Incorrect clustering	Rate of DGD
Correct clustering	22%	42%	64%
Incorrect clustering	1%	35%	36%
Rate of SDG	23%	77%	

Table 7: the correct rate comparison sheet of experiment 3



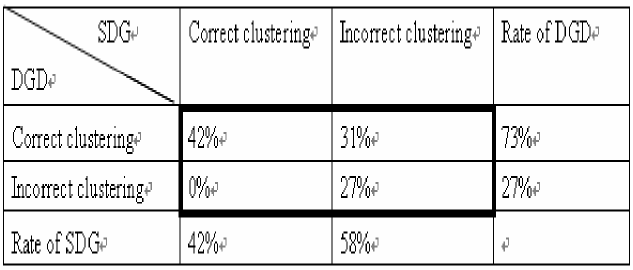
SDG DGD	Correct clustering	Incorrect clustering	Rate of DGD
Correct clustering	16%	44%	60%
Incorrect clustering	1%	39%	40%
Rate of SDG	17%	83%	

Table 8: the correct rate comparison sheet of experiment 4



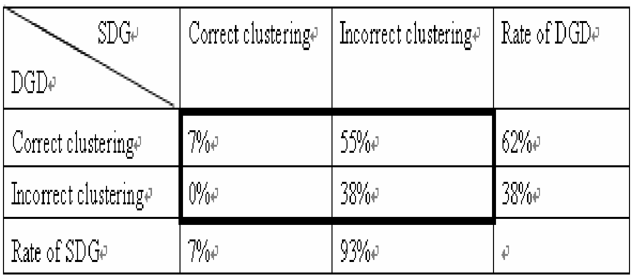
SDG DGD	Correct clustering	Incorrect clustering	Rate of DGD
Correct clustering	72%	9%	81%
Incorrect clustering	4%	15%	19%
Rate of SDG	76%	24%	

Table 9: the correct rate comparison sheet of experiment 5



SDG DGD	Correct clustering	Incorrect clustering	Rate of DGD
Correct clustering	42%	31%	73%
Incorrect clustering	0%	27%	27%
Rate of SDG	42%	58%	

Table 10: the correct rate comparison sheet of experiment 6



SDG DGD	Correct clustering	Incorrect clustering	Rate of DGD
Correct clustering	7%	55%	62%
Incorrect clustering	0%	38%	38%
Rate of SDG	7%	93%	

Table 11: the correct rate comparison sheet of experiment 7

4.2 Discussion

In the DGD algorithm, for each data point α, only those points that are in the same cell of α are considered. The density of each cell is calculated at first. When the total number of data points is n and each dimension, total d dimensions, is divided into m intervals, there will be m^d cells.

The time of checking the density of all cells is k0*[m^d + $(m+1)^d$]. If p(=3^d-1) is the number of nearby cells of one cell, the time of comparing the connected significant cells to generate the two original clustering results is k1*p*[m^d + $(m+1)^d$] at most.

And the time of the cluster revised function $CR()$ is k2*r, where r is the number of $C_{1i} \rightarrow C_{2j}$ and $C_{2j} \rightarrow C_{1i}$ in R_o, $r << m^d << n$.

In the end, the time of checking the cluster's number of all data is k3*n. So the total time complexity is O(m^d)+O(n).

5. Conclusion and Future Work

In this paper, a new grid-based clustering algorithm is called the Deflected Grid-based (DGD) algorithm, which has the obvious wider ranges of size of the cell and threshold of density. And the experimental results verify that the effect of DGD algorithm is less influenced by the size of the cells than other grid-based ones. At the same time, the DGD algorithm still inherits the advantage with the low time complexity.

There are many interesting research problems related to DGD algorithm. One is to find the non-parametric algorithm with the same efficiency of the DGD algorithm at least. And the other is to use algorithm of parallelism to reduce the computational cost.

References:

[1] J. MacQieen. Some methods for classification and analysis of multivariate observation. *Proc. 5th Berkeley Symp. Math. Statist, Prob.*, 1:281-297,1967

[2] L. Kaufman and P.J. Rousseeuw. Finding Groups in Data: *An Introduction to Cluster Analysis*. New York: John Wiley & Sons, 1990.

[3]Charu C. Aggarwal, Philip S. Yu, "An effective and efficient algorithm for high-dimensional outlier detection" *The VLDB journal*, 14:211-221,2005

[4] M. Ester, H. Kriegel, J. Sander, and X. Xu. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", *In Proc. of 2nd Int. Conf. on KDD*, 1996, pages 226-231.

[5] A. Hinneburg and D. A. Keim,. "An Efficient Approach to Clustering in Large Multimedia Databases with Noise", *In Knowledge Discovery and Data Mining*, 1998, pages 58-65.

[6] ANKERST M. etc. "OPTICS: Ordering Points to Identify the Clustering Structure." In Proc. ACM SIGMOD Int. Conf. on MOD, 1999, pages

49-60.

[7] A. H. Pilevar, M. Sukumar, "GCHL: A grid-clustering algorithm for high-dimensional very large spatial data bases", Pattern Recognition Letters 26(2005),999-1010

[8] ZHAO Y.C., SONG J.,"GDILC: A Grid-based Density-Isoline Clustering Algorithm.", *In Proc. Internat. Conf. on Info-net,* Vol 3,pp.140-145,2001 ,

[9]Ma, W.M., Eden, Chow, Tommy, W.S., "A new shifting grid clustering algorithm", *Pattern Recognition* 37 (3),2004,503-514

[10]Alevizos, P., Boutsinas, B., Tasoulis, D., Vrahatis, M.N.,"Improving the K-windows clustering algorithm", *In Proc. 14th IEEE Internat. Conf. on Tools with Artificial Intell*, pp.239-245, 2002.

[11] Wang, Yang, R. Muntz, Wei Wang and Jiong Yang and Richard R. Muntz "STING: A Statistical Information Grid Approach to Spatial Data Mining", *In Proc. of 23rd Int. Conf. on VLDB*, 1997, pages 186-195.

[12] G. Sheikholeslami, S. Chatterjee, and A. Zhang. "WaveCluster: a wavelet-based clustering approach for spatial data in very large databases", *In VLDB Journal: Very Large Data Bases*, 2000, pages 289-304.

[13] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. "Automatic sub-space clustering of high dimensional data for data mining applications", *In Proc. of ACM SIGMOD Int. Conf. MOD*, 1998, pages 94-105.