An Axis-shifted Crossover-Imaged Clustering Algorithm

NANCY P. LIN[1], CHUNG-I CHANG[1], Nien-yi Jan[1],
HAO-EN CHUEH[1], HUNG-JEN CHEN[1,2], WEI-HUA HAO[1]
[1]Department of Computer Science and Information Engineering
Tamkang University
151 Ying-chuan Road Tamsui, Taipei County
TAIWAN
[2]Department of Industrial Engineering and Management
St. John's University,
499, Sec. 4, Tam-King Road, Tamsui, Taipei,
TAIWAN

nancylin@mail.tku.edu.tw, taftdc@mail.tku.edu.tw, yijan@cht.com.tw,
890190134@s90.tku.edu.tw, chenhj@mail.sju.edu.tw, 889190111@s89.tku.edu.tw
http://mail.tku.edu.tw/125502/csie/pclin.htm http://mail.tku.edu.tw/taftdc/defaul.htm

Abstract: - With low computation time, the grid-based clustering algorithms are efficient clustering algorithms, but the size of the predefined grids and the threshold of the significant cells are seriously influenced their effects. In grid-based clustering system, the data space is partitioned into a finite number of cells to form a grid structure and then performs all clustering operations on this obtained grid structure. The ADCC algorithm is the first one to use axis-shifted strategy to reduce the influences of the size of the cells and inherits the advantage with the low time complexity. But it still uses the cell-clustering twice, the Axis-shifted Crossover-Imaged Clustering Algorithm, called ACICA$^+$, is proposed to use cell-clustering only once and still has the same results. The main idea of ACICA$^+$ algorithm is to shift the original axis in each dimension of the data space after the image of significant cells generated from the original grid structure have been obtained. Because the shifted grid structure can be considered a dynamic adjustment of the size of original cells and the threshold of significant cells, the new image generated from this shifted grid structure will be used to revise and replace the originally obtained significant cells. Finally the clusters will be generated from this crossover image. The experimental results verify that, indeed, the effect of ACICA$^+$ algorithm is less influenced by the size of the cells than other grid-based algorithms. Finally, we will verify by experiment that the results of our proposed ACICA$^+$ algorithm outperforms than others.

Key-Words: - Data Mining, Grid Structure, Crossover Image, Cell, Shifted Grid, Cater-Corner Significant Cell

1 Introduction

Up to now, many clustering algorithms have been proposed [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14], 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], CLIQUE [13], and ADCC [14].

In general, grid-based clustering algorithm is the most computationally efficient algorithm, but the effect of grid-based clustering algorithm is 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 Axis-shifted Crossover-Imaged Clustering (ACICA) algorithm in this paper.

The main idea of our proposed ACICA is to utilize some predefined grids and a predefined threshold to identify the image of significant cells in the first grid structure. Then, the modified grids which are deflected to half size of the grid are used to identify the image of significant cells in the second grid structure again. Next, the two images of significant cells are crossover to generate the final clustering result.

The rest of the paper is organized as follows: In section 2, some popular grid-based clustering algorithms are mentioned again. In section 3, our proposed clustering algorithm, ACICA algorithm, is introduced. In section 4, an experiment and some discussions are displayed. Section 5 is the conclusion.

2 Grid-based Clustering Algorithm

Grid-based clustering algorithm is an efficient clustering algorithm, and three famous grid-based clustering algorithms are STING [11], CLIQUE [13], and ADCC [14].

STING (Statistical Information Grid-based algorithm) (Wang et al., 1997) is a grid-based clustering technique. It employs a hierarchical structure of grid cells and uses longitude and latitude to divide the spatial space into rectangular grid cells. Each cell at a high level is partitioned to form a number of cells at the next lower level. Statistical information regarding the attributes in each grid cell (such as the number of data, mean, maximum, minimum, and distribution values) is pre-computed and stored. These statistical parameters are useful for query processing. At first, it selects a layer to begin with. Then, for each cell of this layer, labels the cell as relevant if its confidence interval of probability is higher than the threshold. Next, it goes down the hierarchy structure by one level and goes back to check those cells is relevant or not until the bottom level. Afterwards, return those regions that meet the requirement of the query. And finally, to retrieve those data fall into the relevant cells.

The CLIQUE (Clustering In QUEst) (Agrawal et al., 1998) clustering algorithm integrates density-based and grid-based clustering. For high dimensional data sets, it provides automatic sub-space clustering of high dimensional data. It consists of the following steps: First, to uses a bottom-up algorithm to find dense units in different subspaces. The CLIQUE based on the Apriori property that if a k-dimensional unit is dense, then so are its projections in (k-1)-dimensional space. Second, it uses 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.

The idea of ADCC (Adaptable Deflect and Conquer Clustering) (Lin et al., 2007) is to utilize the predefined grids and predefined threshold to identify the significant cells, by which nearby cells that are also significant can be merged to develop a cluster in the first place. Next, the modified grids which are deflected to half size of the grid are used to identify the clusters again. Finally, the new generated clusters and the initial clusters are merged to be the final clustering result.

In fact, to reduce the influences of the size of the predefined grids and the threshold of the significant cells, a new grid-based clustering algorithm proposed is called Axis-shifted Crossover-Imaged Clustering Algorithm ($ACICA^+$) in this paper.

3 The Axis-shifted Crossover-Imaged Clustering Algorithm

The Axis-shifted Crossover-Imaged Clustering ($ACICA^+$) algorithm further shifts the grid structure by half a cell width in each dimension, the same as ADCC algorithm, but ADCC generates the final clustering by using clustering procedure three times. To improve the chief shift, the axis-shifted crossover-imaged clustering is proposed. The $ACICA^+$ generates the final clustering results only using one clustering and no more memory.

3.1 ADCC Algorithm

After the grid structure is built and the first clustering result is generated, the ADCC shifts the coordinate axis by half a cell width in each dimension and has the new grid structure and the second clustering result is generated. Then the two sets of clusters are combined into the final result. The procedure of ADCC is shown in the following steps in fig.1.

1) Generate the first grid structure.
2) Identify the significant cells.
3) Generate the first set of clusters.
4) Transform the grid structure.

5) Generate the second set of new clusters.

6) Revise the original clusters.

7) Generate the clustering result.

Fig.1 procedure of ADCC

Step 1: Generate the first 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 first grid structure.

Step 2: Identify significant cells.
Next, the density of each cell is calculated to find out the significant cells who se 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 clusters is denoted as S_1.

Step 4: Transform the grid structure.
The original coordinate origin is next shifted by distance d in each dimension of the data space, so that the coordinate of each point becomes d less in each dimension.

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 transformed grid structure. The set of new clusters generated here is denoted as S_2.

Step 6: Revise original clusters.
The clusters generated from the second grid structure can be used to revise the originally obtained clusters. And the first grid structure can also be used to revise the second obtained clusters. The procedure of Revision of the original clusters is shown in fig.2.

1) Find each overlapped cluster C_{2j} for $C_{1i} \in S_1$

2) Generate the rule $C_{1i} \rightarrow C_{2j}$,
where $C_{1i} \bigcap C_{2j} \neq \phi$, $C_{2j} \in S_2$.

3) Find each overlapped cluster C_{1i} for $C_{2j} \in S_2$

4) Generate the rule $C_{2j} \rightarrow C_{1i}$,
where $C_{2j} \bigcap C_{1i} \neq \phi$.

5) $R_o = \{C_{1i} \rightarrow C_{2j} \mid C_{1i} \bigcap C_{2j} \neq \phi\} \bigcup \{C_{2j} \rightarrow C_{1i} \mid C_{2j} \bigcap C_{1i} \neq \phi\}$

6) $C_{1i} \in S_1$ are revised by using the cluster revised function $CM()$.

Fig.2 the sketch of Revision of the original clusters

Step 6a: Find each overlapped cluster C_{2j} for $C_{1i} \in S_1$, and generate the rule $C_{1i} \rightarrow C_{2j}$, where $C_{1i} \bigcap C_{2j} \neq \phi$, $C_{2j} \in S_2$. 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 $C_{2j} \bigcap C_{1i} \neq \phi$.

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 $CM()$. The cluster modified function $CM()$ is shown in fig.3.

```
For each C1i ∈ S1
  Let X' := X;
  Repeat
    oldX' := X';
    For each Y→Z in R0 Do
      If Y ⊂ X' then
        X' := X' ∪ Z;
        If Z ∈ S1 then
          S1 := S1 – {Z};
        Else
          S2 := S2 – {Z};
        Endif
      Endif
  Until ( oldX' = X' );
  C1i := X';
End
S1 := S1 ∪ S2
```

Fig.3 the CM algorithm

Step 7: Generate the clustering result.
After all clusters of S_1 have been revised, S_2 is

the rest of the original set of S_2 after revision. The final set of clusters is $S_1 = S_1 \cup S_2$. The result will be the same as S_2 revised by S_1.

3.2 ADCC Example

In fig.4, the two-dimensional example with 600 points is easy to be divided into two clusters. The example goes through the ADCC algorithm.

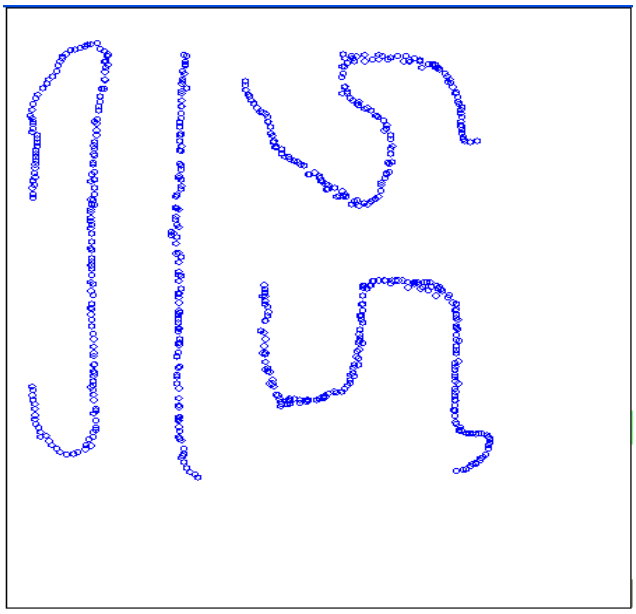


Fig.4 two-dimensional example

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.5.

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, here the threshold is 4.

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. 6.

step4: Deflect the grid structure.

The original grid structure is deflected by distance d in each dimension of the data space.

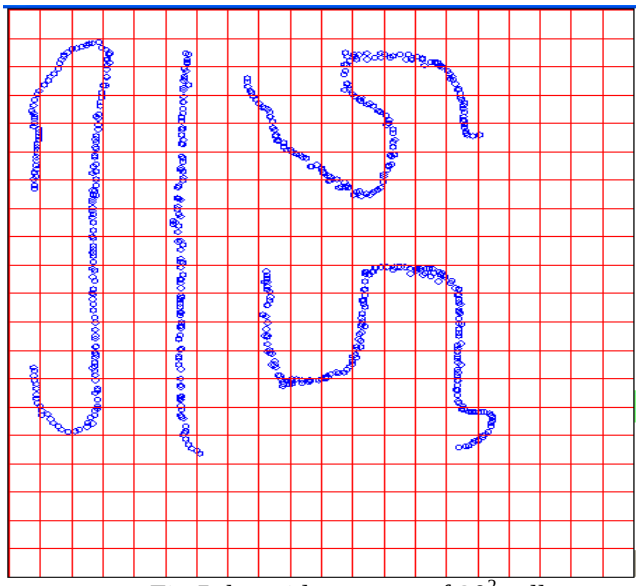


Fig.5 the grid structure of 20^2 cells

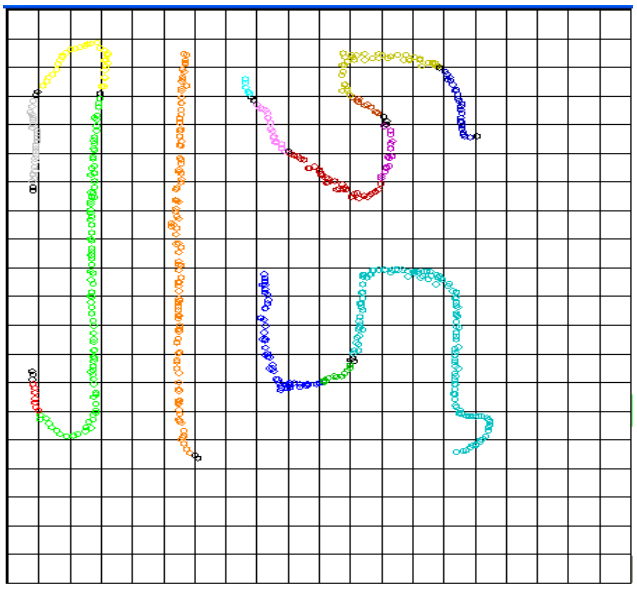


Fig.6 result of first clustering

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. 7.

Step 5: Generate the set of new clusters.

Here, the cell density of new grid structure is calculated. 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 $S_2 = \{C_{21}, C_{22}, \ldots, C_{14}\}$, shown in fig. 8.

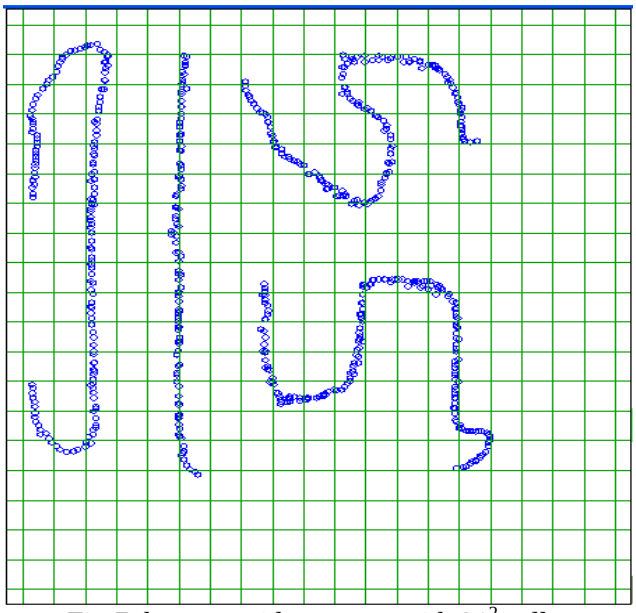


Fig.7 the new grid structure with 21^2 cells

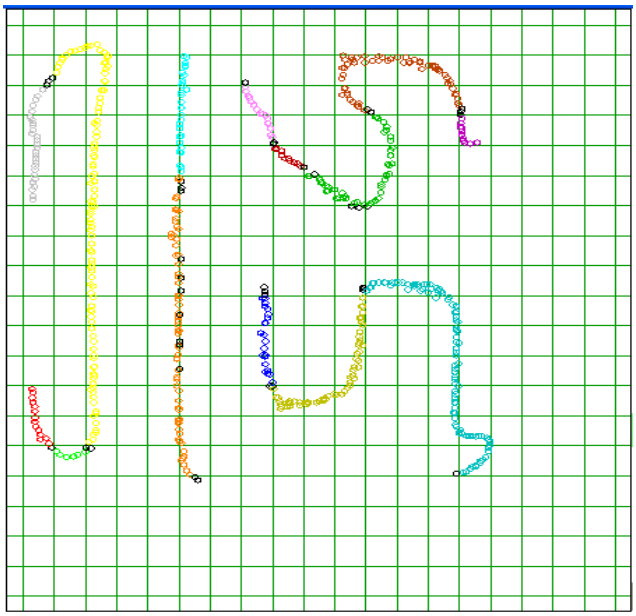


Fig.8 Result of the second clustering

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. 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.

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

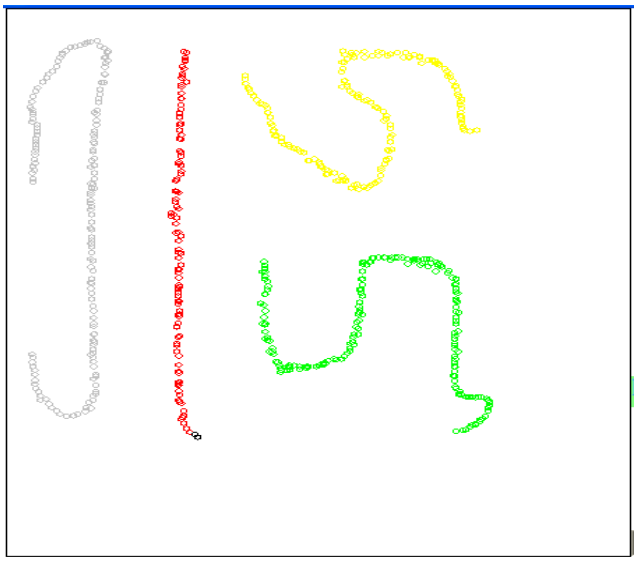


Fig.9 the final clustering result

3.3 ACICA$^+$ Algorithm

After the two grid structures are built, the ACICA$^+$ projects the second image of significant cells on the original grid structure to be the crossover image. Finally the clusters will be generated from this crossover image. The procedure of ACICA$^+$ is shown in the following steps.

Step 1: Generate the first 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 G1.

Step 2: Identify the significant cells.

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

Step 3: Transform the new grid structure.

The original grid structure is next shifted by distance d in each dimension of the data space.

Step 4: Identify significant cells.

Next, the density of each cell is calculated to find out the new image of significant cells whose densities exceed a predefined threshold.

Step 5: Generate the crossover image and the pre-clustering results.

While $SC_1 \neq \phi$ do
  χ = new $\chi \in SC_1$
  $X = \{\chi\}$
  $SC_1 = SC_1 - \chi$
  $Cluster_k = CEXT_\chi$
  While $y = Neighbor(X) \in SC_1$ then
    $Cluster_k = Cluster_k \cup CEXT_y$
    $SC_1 = SC_1 - y$
    $X = X \cup \{y\}$
  Next y
Next $K = K + 1$

※ $CEXT_i = \{C_{1i}\} \cup \{C_{2j} \mid C_{1i} \cap C_{2j} \neq \phi, C_{2j} \in SC_2\}$

The second image of significant cells is generated and projected on the original grid structure to be the crossover image. The significant cell in first grid structure and its crossover image are combined to be the extent of significant cell (CEXT).

Step 6: Generate the clustering result.

if $CEXT_\Omega - \{C_{1\Omega}\} \in Cluster_p \cap .. \cap Cluster_s$ then
  $Cluster_p = Cluster_p \cup .. \cup Cluster_s$
  $Cluster_p .. Cluster_s = \phi$

Find each $OC^1_{X_1..Xn}$, the set of overlapped significant cells $C^2{}_{X1..Xn}$ for $C^1{}_{X1..Xn} \in G1$, grid structure 1, where $C^2{}_{X1..Xn}$ is the significant cell in the second n-dimensional grid structure. $OC^1_{X_1..Xn} = \{C^2{}_{X1..Xn} \mid C^2{}_{X1..Xn} \bigcap C^1{}_{x1..xn} \neq \phi\}$, where $C^2{}_{X1..Xn}$ is significant cells in G2, the second grid structure. By using the crossover image, the nearby significant cells in the original grid structure are clustered with their extent of significant cells. The clusters are combined into one cluster depending on the cater-corner significant cells, shown in fig.10, which are the significant cells in second grid structure. The cater-corner significant cells are belongs to at least two clusters in the original grid structure. Then the set of clusters are combined into the same cluster.

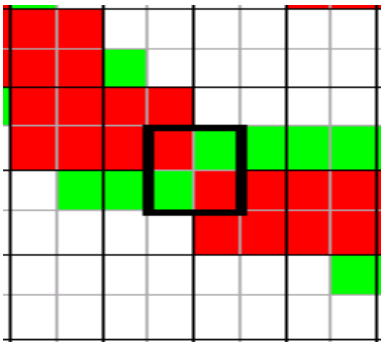


Fig.10 cater-corner significant cell

3.4 ACICA$^+$ Example

In this place, the two-dimensional example, as shown in fig.11, with 799 points is easy to be divided into four clusters. The example goes through the algorithm.

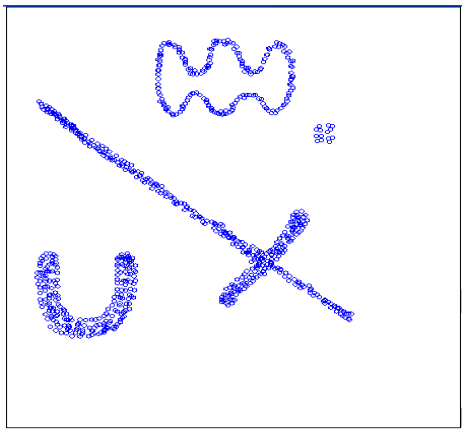


Fig.11 figure of example

Step 1: Generate the first 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.12.

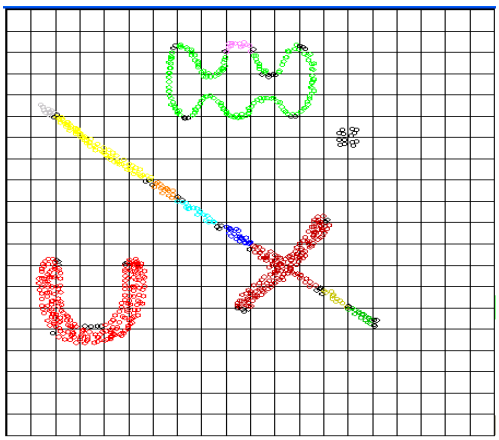


Fig.12 the grid structure G1

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, here the threshold is 7. And the image of significant cells is shown in fig.13.

Step3: 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. 14.

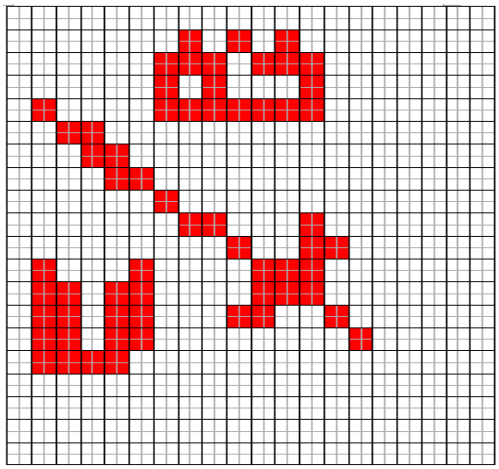


Fig.13 image of significant cells

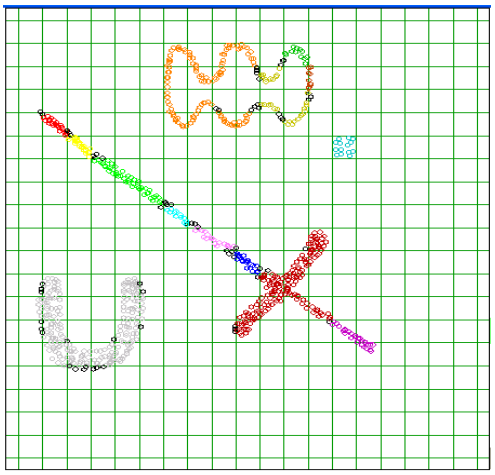


Fig.14 second grid structure G2

Step 4: Identify significant cells.

Here, the density of each cell is calculated to find out the significant cells whose densities exceed the same predefined threshold. And the image of significant cells is shown in fig.15.

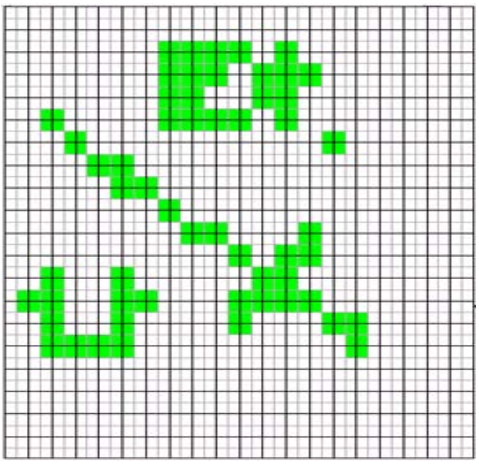


Fig.15 image of significant cells

Step 5: Generate the crossover image and the pre-clustering results.

The second image, shown in fig.9, of significant cells is projected on the original grid structure to be the crossover image shown in fig.16. And the pre-clustering result is shown in fig.17.

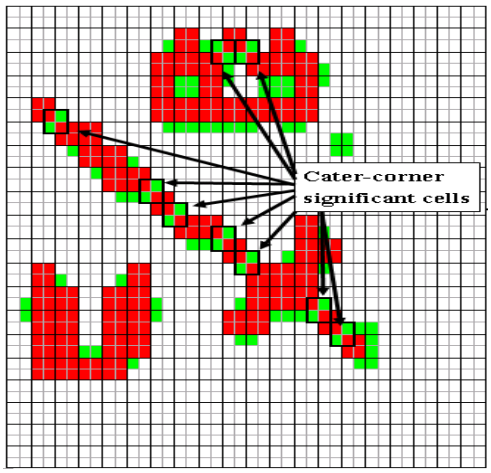


Fig.16 crossover image of significant cells

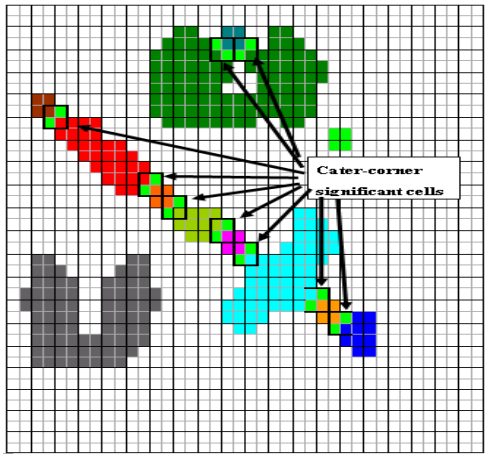


Fig.17 pre-clustering result

Step 6: Generate the clustering result.

By using the crossover image, the nearby significant cells in the original grid structure are clustered with their extent of significant cells, shown in fig.17. If the cater-corner significant cells are belongs to at least two clusters in the original grid structure then the set of clusters are combined into the same cluster. The final clustering result is shown in fig.18.

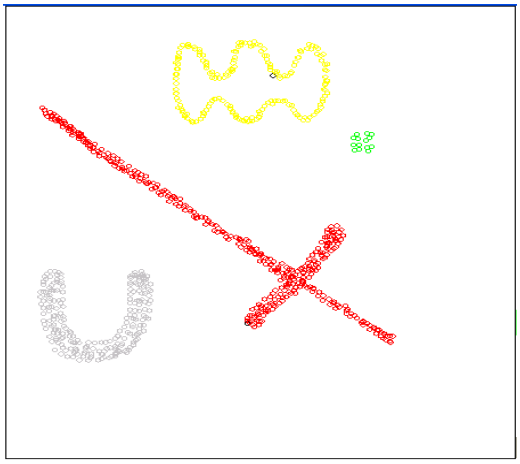


Fig.18 the clustering result

4. Experiments and Discussions

Here, we experiment with seven different data shown in fig.19 ~ fig.25. And the features are presented 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.19 experiment 1

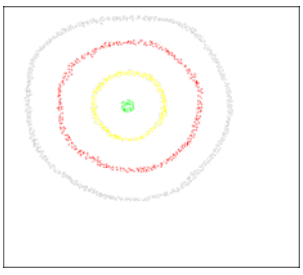


Fig.20 experiment 2

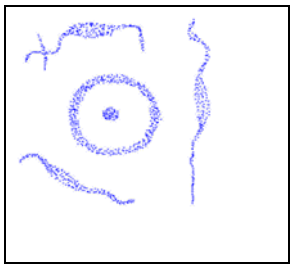


Fig.21 experiment 3

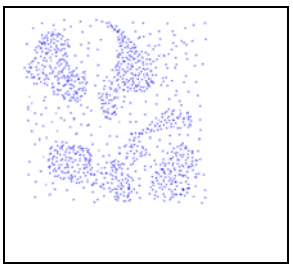


Fig.22 experiment 4

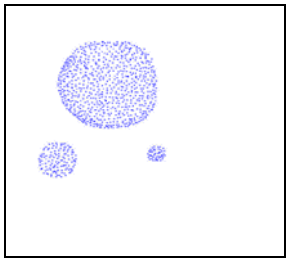


Fig.23 experiment 5

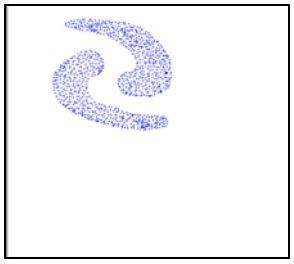


Fig.24 experiment 6

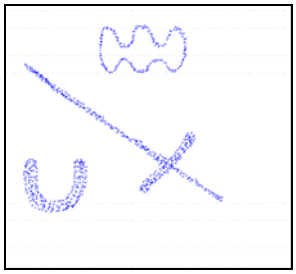
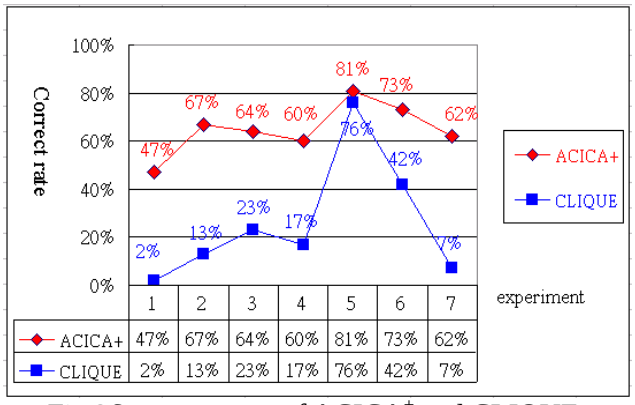


Fig.25 experiment 7

4.1 . Experiment

Fig.26 shows the correct rates of ACICA$^+$ and CLIQUE, where the correct clustering result of CLIQUE is by using one of original or new grid structures in the experiment. The correct rates of ACICA$^+$ are all higher than CLIQUE. In the experiment, 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).



	1	2	3	4	5	6	7
ACICA+	47%	67%	64%	60%	81%	73%	62%
CLIQUE	2%	13%	23%	17%	76%	42%	7%

Fig.26 correct rates of ACICA$^+$ and CLIQUE

4.2 Discussion

In the ACICA$^+$ algorithm, the density of each cell is calculated. When the total number of data is n and each dimension, total d dimensions, is divided into m intervals, there will be m^d cells. The time of allocating the data and checking the density of all cells is k0*n. If p(=2d) is the number of nearby cells of one cell, the time of checking if the cell significant is k1*p*[m^d + $(m+1)^d$] at most. If the number of significant cells in G1 and G2 are s1 (<<m^d) and s2 (<< $(m+1)^d$). Then the time to find the extent of significant cell in G1 is k2*s2. The time of clustering in ADCC is k3*[s1 + s2 + r], where r is the number of rules used in revised function *CM*() , but the time of clustering in ACICA$^+$ is only k4*(s1+ c) < k3*[s1 + s2 + r], where c is the number of cater-corner significant cells and c << s2 << m^d. In the end, the time of checking the cluster's number of all data is k5*n. So, the time of clustering by using ACICA$^+$ is shorter than using ADCC.

5. Conclusion and Future Work

The algorithm we proposed, the Axis-shifted Crossover-Imaged Clustering Algorithm (ACICA$^+$), makes important contribution to support the obvious wider ranges of size of the cell and threshold of the density to reduce the drawbacks of grid-based clustering algorithms. We also discuss the details and advantages of ACICA$^+$ to compare with ADCC, which is the first grid-based algorithm to support the obvious wider ranges of size of the cell and threshold of the density. And the clustering results of ACICA$^+$ are exactly the same as the results of ADCC. From the present results obtained, owing to the number of cell-clustering times, the ACICA$^+$ has only once but ADCC twice. So it is fast and simple to realize that the ACICA$^+$ algorithm not only inherits the advantage with the low time complexity, but also is verified by experiment that it outperforms than ADCC.

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. Internet. 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.

[14] N. Lin, C. Chang, and C. Pan. "An Adaptive Deflect and Conquer Clustering algorithm", *In Proc. of 6th WSEAS Int. Conf. ACOS'07*, 2007, pages 156-160.