# An Adaptable Deflect and Conquer Clustering Algorithm

NANCY P. LIN, CHUNG-I CHANG, CHAO-LUNG PAN
Department of Computer Science and Information Engineering
Tamkang University
151 Ying-chuan Road Tamsui, Taipei County
TAIWAN

*Abstract*: - The grid-based clustering algorithm is an efficient clustering algorithm, but the effect of the algorithm is seriously influenced by the size of the predefined grids and the threshold of the significant cells. Thus, in this paper, to reduce the influences of the size of the predefined grids and the threshold of the significant cells, we adopt deflect and conquer techniques to propose a new grid-based clustering algorithm, which is called Adaptable Deflect and Conquer Clustering (ADCC) algorithm. The idea of ADCC 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 significant cells again. Finally, the new generated significant cells and the initial significant cells are merged so as to offset the round-off error and improve the precision of clustering task. And we verify by experiment that the performance of our new grid-based clustering algorithm, ADCC, is good.

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

## 1 Introduction

Clustering analysis is an important method of data mining. The goal of clustering analysis is to group the data objects into clusters according to two critical rules: maximizes the intra-cluster similarity, and minimize the inter-cluster similarity.

Many clustering algorithms have been proposed up to now, and those algorithms can be categorized into partitioning-based, density-based, grid-based, and so on. In partition-based clustering algorithms, such as k-means and k-medoids, some reallocation techniques are used to improve the partitioning result by moving data objects from one cluster to another. Those tend to find clusters with spherical shape and similar size. In density-based clustering algorithms, such as DBSCAN [1], DENCLUE [2] and OPTICS [3], makes use of the density of data points within a region to form clusters. It can find the clusters of arbitrary shapes and handle noise easily. Also it has the feature of only one scan, but it needs the density parameters as the termination condition. In grid-based clustering algorithms, such as STING [4], WaveCluster [5] and CLIQUE [6], it considers cells rather than data points. The grid-based clustering algorithms approximate the dense regions of the data space by dividing it into a finite number of cells and identifying dense cells that contains the number of data points more than the specific threshold. Clusters are then formed by connecting these dense cells.

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 Adaptable Deflect and Conquer Clustering (ADCC) algorithm in this paper.

The main idea of our proposed ADCC algorithm is to utilize some predefined grids and a predefined threshold to identify the first generation significant cells. Then, the modified grids which are deflected to half size of the grid are used to identify the second generation significant cells again. Next, the two generations are merged to come into being the final clusters. The purpose of the merge phase is to offset the round-off error and improve the precision of clustering task.

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, ADCC 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 two famous grid-based clustering algorithms are STING [4] and CLIQUE [6].

STING (Statistical Information Grid-based algorithm) (Wang et al., 1997) exploits the clustering properties of index structures. It employs a hierarchical structure of grid cells and uses longitude and latitude to divide the spatial area into rectangular grid cells. At first, it selects a layer to begin with. 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. Finally, to retrieve those data fell 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 Adaptable Deflect and Conquer Clustering (ADCC) algorithm in this paper.

## 3 Adaptable Deflect and Conquer Clustering Algorithm

The Adaptable Deflect and Conquer Clustering (ADCC) algorithm further deflects the cell margins by half a cell width in each dimension and has the second result of using the SGDC algorithm then combines the two sets of clusters into one result.

The ADCC algorithm can not only save the data points lost in the partition related to the SGDC, but also improve the clustering result that decreases the effect of corners of cell, weakness of grid-based algorithms, to combine the related cells to the same cluster. The ADCC algorithm is illustrated in Fig.1.

step1.Partition the spatial data space into non-overlapping hyper-rectangles (first generation).
step2.Calculate all cells' density.
step3.Check whether the cell is significant or not. If the number of data points in one cell exceeds the density threshold then the cell is significant.
step4.Combine the connected significant cells to be one cluster. That is, significant cells connected with at least one edge with others are combined into one cluster.
step5.The result is the first generation set of clusters S1.
step6.Repartition the data space by deflecting half a cell in each dimension to have the new non-overlapping hyper-rectangles (second generation).
step7.Repeat the step 2 to step 5 to get the result of second generation set of cluster S2.
step8.Merge the S1 and S2 to have the final result by DeflectAndConquer procedure.

DeflectAndConquer procedure:

step1: Check all significant cells in the two phases by checking the first and second phase recursively.
step2: If all significant cells are checked then stop.
step3: If the significant cell and all its significant neighboring ones are not belong to any set then set the cells to a new cluster and go back to step1 and check the next significant cell.
step4: Check whether the significant cell and all its neighboring significant ones belong to one cluster or not. If not, the cell and all its neighboring significant cells are set to the same cluster.
step5: If the significant cell and its cater-corner significant neighboring ones belong to different clusters then change the aim to check the other phase. We check the cell which covers parts of the two significant cells in the previous phase.
step6: If it is significant then the two significant cells and all their neighboring significant ones in the previous phase are combined into the same cluster. Otherwise, they belong to two different clusters.
step7:Go back to check the previous phase unceasingly.

```
Algorithm ADCC
Begin
  Call SGDC to return the first-phase clustering result
  Repartition the data space by deflecting the margin of all cells in each dimension by half the
    length of a cell
  Call SGDC to return the first-phase clustering result
  Call DeflectAndConquer to integrate the final result
End

Algorithm SGDC
Begin
  Partition the spatial data space into non-overlapping hyper-rectangles.
  //to check the cells are significant or not//
  For all cells
      calculated each cell's density
      cells with higher density than the threshold are the significant ones.
  //to cluster the significant cells//
  For all cells
    if the cell is significant
    then check its neighboring significant cells
    if all its neighboring significant cells are not belonged to any cluster
    then all belong to a new cluster
    elsif some of its neighboring significant cells are belonged to the same one cluster
    then all belong to the cluster
    elsif all its neighboring significant cells are belonged to at least two cluster
    then combine the cells belonged to the clusters into one single cluster
  endfor
End

Algorithm DeflectAndConquer
Begin
  for all significant cells in the two phases, check the first phase and second phase recursively
    if all significant cells are checked then stop
    if the significant cell and all its significant neighboring cells are not belong to any set
    then set the cells to a new cluster and go back to the for loop to check the next
        significant cell
    endif
    while any significant cell and its neighboring significant cells belongs to a specific set of
      clusters then the cell and all its neighboring significant cells are set to the same cluster
    endwhile
    if the significant cell and its cater-corner significant neighboring cell(s) are belong to
      different clusters
    then change to another phase, we check the cell which covers parts of the two significant
      cells
        if it is significant
        then the two significant cells in the previous phase are combined into same cluster
        else they are belong to two different clusters
        endif
        go back to check the previous phase unceasingly
    endif
  endfor
  return the final result
End
```

Fig.1 the ADCC algorithm

### 3.1 Example

In this place, the two dimensional example with 1032 points is easy to be divided into two clusters. At first, the spatial data space is partitioned into non-overlapping hyper-rectangles, and each cell's density is calculated. In the example, the two dimensional spatial space is partitioned into 20 x 20 cells and the density threshold is 10. Confirm the cells whose density above the threshold is the significant cell. The neighboring significant cells are merged into the same cluster, the result as showed in Fig.2. Then, Repartition the data space by deflecting half a cell in each dimension to have the new non-overlapping hyper-rectangles. At this time, the two dimensional spatial space will be partitioned into 21*21 cells, the size of marginal cells is a half of cell and the size of corner cells is a quarter of cell, and the density threshold is 10.

Again, to confirm all cells if anyone's density is above the threshold then the cell is significant. If the neighboring significant cells are merged into the same cluster, the result will be showed in Fig.3.

Then, do the procedure of DeflectAndConquer to improve the weakness of Fig.2 and Fig.3, that black colored cells in the marginal of the cluster will be recovered to be significant cells. The final result of using ADCC is showed in Fig.4; there are only 2 clusters and some sparse outliers.

If the spatial space is partitioned to 40 x 40 cells and the density threshold is changed into 5, the clustering result is showed in Fig.5; There are 12 clusters and many sparse outliers in Fig.5, but only 2 clusters in Fig. 4. It means that even the size of cell is narrowed and the threshold is reduced simultaneously, the result is not as good as one of using the ADCC algorithm.

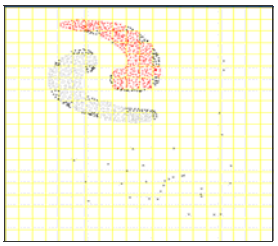

Fig.2

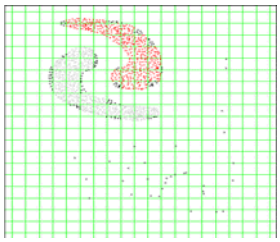

Fig.3

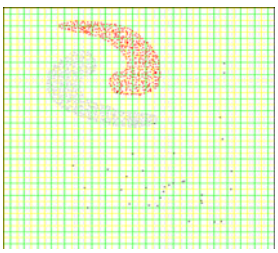

Fig.4

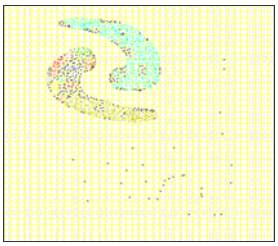

Fig.5

# 4 Experiments and Discussions

Here, we experiment with two different data. The features are showed as Table 1.

| Data number | Data quantity | Distribution type | Natural clustering number |
|---|---|---|---|
| Fig.6 | 600 | Four separate linear data | 4 |
| Fig.7 | 1100 | Four concentric circular | 4 |

Table 1 experimental data feature

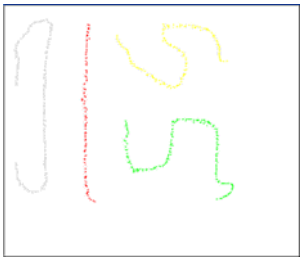

Fig.6 experiment 1

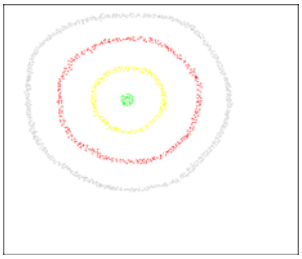

Fig.7 experiment 2

| No. of cells in each direction | No. of threshold of density | |
|---|---|---|
| | CLIQUE | ADCC |
| 1~18 | 0 | 0 |
| 19 | 0 | 1~4 |
| 20 | 1 | 1~5 |
| 21 | 1 | 1~5 |
| 22~24 | 0 | 1~3 |
| 25~26 | 1 | 1~3 |
| 27~30 | 0 | 1~3 |
| 31~38 | 0 | 1~2 |
| 39~70 | 0 | 1 |

Table 2: Result of Experiment 1

## 4.1 Experiment 1

The experiment 1, four separate linear curves, is composed of 600 data points as shown in Fig.6. The feature is showed in Table 1. The experimental Result is showed in Table 2. For CLIQUE, there is only 4 sets of (No. of cells, threshold) are able to find the correct sets of clusters. They are (20, 1) ,(21, 1) , (25, 1) ,(26, 1). When we choose other sets of (No. of cells, threshold), the data points in experment 1 will be set to more than 4 clusters and there will be many outliers. But the ADCC provides up to 89 sets of (No. of cells, threshold) to find the correct sets of clusters. If we only check the number of cells in one dimension, there are only 4 selections in CLIQUE can find the correct sets of cluster, but ADCC provides up to 52 selections. And check the fitting range of threshold of density in one cell, only density=1 is the unique choice in CLIQUE, but the possible choices of threshold of density are 1 to 5 in ADCC.

## 4.2 Experiment 2

The experiment 2, four concentric circular curves, is composed of 600 data points as shown in Fig.6. The feature is showed in Table 1. The experimental Result is showed in Table 3. For CLIQUE, there is still only 32 sets of (No. of cells, threshold) and the threshold of density only 1 is the unique choice to find the correct sets of cluster. But the ADCC provides up to 146 sets of (No. of cells, threshold) and the possible choices of threshold of density are from 1 to 8.

So, when we use the ADCC algorithm, it's easy to reduce the influences of the size of the predefined grids and the threshold of the significant cells to get the right results of clustering.

| No. of cells in each direction | No. of threshold of density | |
|---|---|---|
| | CLIQUE | ADCC |
| 1~18 | 0 | 0 |
| 19 | 0 | 8 |
| 20~23 | 0 | 4~7 |
| 24~30 | 1 | 1~5 |
| 31~34 | 1 | 1~4 |
| 35~41 | 1 | 1~3 |
| 42~47 | 1 | 1~2 |
| 48 | 1 | 1 |
| 49~50 | 0 | 1~2 |
| 51~57 | 1 | 1~2 |
| 58~87 | 0 | 1 |

Table 3: Result of Experiment 2

## 4.3 Discussion

In the ADCC algorithm, for each data sample α, only those samples that are in the same cell of α are considered. The density of such cell is calculated. When the number of data samples 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 checking the cell is significant or not is $k1*p*[m^d + (m+1)^d]$ at most. So the time of DeflectAandConquer in ADCC is $k2*[m^d + (m+1)^d]$. 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, we introduced an adaptable deflect and conquer grid clustering algorithm, ADCC algorithm, which has the obvious wider ranges of size of the cell and threshold of density and improves the precision of clustering task. At the same time, the ADCC algorithm still inherits the advantage with the low time complexity.

There are many interesting research problems related to ADCC algorithm. One of the most interesting problems for future research is how to find the non-parametric algorithm with the same efficiency of the ADCC algorithm at least.

*References:*

[1] 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.

[2] 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.

[3] ANKERST M. etc. "OPTICS: Ordering Points to Identify the Clustering Structure." *In Proc. ACM SIGMOD Int. Conf. on MOD*, 1999, pages 49-60.

[4] 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.

[5] 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.

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

[7] Z. Huang, "Extensions to the k-means algorithm for clustering large data sets", *Data Mining and Knowledge Discovery*, 1998, vol. 2, pages 283-304.