CHAPTER 5

CLIQUE OPTIMIZATION ALGORITHM

5.1 INTRODUCTION

Clustering is one among the foremost helpful tasks in data mining method which might partition objects of a data set into distinct clusters such that two objects from one cluster are almost like one another, whereas two objects from distinct clusters are not. Spatial clustering, progressing to establish clusters, or densely populated areas in exceedingly massive spatial data sets, is a vital task of spatial knowledge mining.

The spatial information area is split by hyper-planes that are entertained with axis-paralleled histogram in the CLIQUE Optimization algorithm. A division of the data space area depends on the natural distributing character of the input data area to enhance the accuracy and potency of spatial clustering. Simultaneity, the outstanding distinction between density-region and sparse-region makes the setting of density threshold parameter simply and reduces the parameter dependence of spatial clustering algorithm.

The existing CLIQUE algorithms have some limitations for dividing the sub space data into hyper rectangular cells in same width and pruning technique which may solve heap of your time operations primarily based on debasing, additionally threshold worth will have a significant half
for solving the region division. The density threshold value is set by the user with immoderate value.

5.2 THE CLIQUE OPTIMIZATION ALGORITHM

The optimized-division is a technique which might divide the input space betting on the distributing of spatial data automatically and secure the minimize variety of divided areas that repeat the particular distributing character of spatial data. The optimized-division of CLIQUE Optimization algorithm operates along with the following steps.

**Step1:** Building the 2-dimensional coordinates of data space and projecting the purpose of each data point on 2 axes respectively;

**Step2:** Creating the axis-parallel histogram along the directions of the axes. Setting the unit of coordinate as $a=1$, it describes the algorithm simply.

**A standard significance test used in CLIQUE Optimization algorithms**

$$X^2 = \frac{2(V_{\text{val}} - V_{\text{exp}})^2}{V_{\text{exp}}} \geq Z_a^2$$

In the formula 1 $V_{\text{val}}$ is the value of the histogram, $V_{\text{exp}}$ is the value of expect the zone area expecting points and $Z_a$ is the value of zone area setting coordinate.

5.2.1 Description of the Algorithm

In this study, the CLIQUE Optimization algorithm follows the major steps.
**Step 1:** divide the space with X, Y axis depending on distribution of input Spatial data to avoid demolishing of accepted clusters.

**Step 2:** Apply of pruning concept to find Density area and sparse area.

**Step 3:** To find the Threshold Parameter (P); T→minimum number of data Points in density area.

**Definition 1**

**Density threshold (T)**

Let the density threshold be T. It is the minimum number of data points in unit space in the density-region.

**Definition 2**

**Density-area and spare-area**

If the number of data points in this region is not less than the threshold, the region is defined to be dense-area. Otherwise, the area is defined to be sparse-area.

**Algorithm**

**Input:** Crime Spatial datasets D, T, Points of X, Y

**Output:** Cluster area for the given D

**Step1:** To find the particular crime data in the given input crime datasets.
**Figure 5.1 Given Spatial Data Space**

**Step 2:** Find the X, Y coordinates of data space, Additionally X, Y coordinate points projected the input data space and the data point respectively. The proposed algorithm count number of projecting points in every gap on X, Y coordinates. According to the data points, the proposed algorithm constructs scale points for every section of X, Y coordinates.

**Figure 5.2 Finding X, Y Axes**
Step 3: Apply Optimization and find the max out cluster in the data space.

In this step algorithm used climbing method to find the best splitting points. Using this method it finds the cluster area between two density regions, which can assure the statistically important region. And the splitting pairs can be determined along with the direction of X, Y coordinates paralleled.

Step 4: To identify the Dense Area clusters and Sparse Area using Threshold parameter (T)
In this step, if the count the number of data points in every area is N, recording the value is NUMi, calculating the density of Area with the x, y coordinate value and T ≥ NUMi, it is Density of Cluster, otherwise it is Sparse area.

Figure 5.5 Identify Dense Clusters and Sparse Areas based on T

**Step 5:** End

5.2.2 Clique Optimization Algorithm Pseudo Code

Figure 5.7 shows the flowchart of clique optimization algorithm how the flow of data taken over to find the hotspot.

**Table 5.1 CLIQUE Optimization Algorithm Pseudo Code**

```
Begin
Assign the crime data base DB
Produce KDE map based in DB
To find the X, Y coordinates which point to the cluster area
Optimize (cluster area, Scale points)
Threshold Value T= difference (Cluster area, sparse area)
If (T > Numi)
Hot spot area
Else
Repeat Optimize
End
```
5.3 CLIQUE OPTIMIZATION IMPLEMENTATION

Introduction

Clustering is one with all the essential data mining issues. Generally, the goal in clustering is, given a dataset, to seek out naturally occurring groups among the datasets, or areas within the space generated by the datasets where the density of data points is higher than the normally expected one. Figure 5.6 shows an example of a 2 dimensional datasets. One will naturally establish two dense areas within the figure that is a must to invent clusters.

It is natural to approach this problem by computing the similarities between individual information points within the data set and partitioning the data points during an approach that yields groups where the similarity of points among every partition is as high as feasible, whereas points in several partitions are as dissimilar as feasible.

![Subspace Clustering Example](image)

**Figure 5.6 Subspace Clustering Example**
Flow Chart for clique optimization Algorithm

Figure 5.7 CLIQUE Optimization Algorithm Flow Chart
Issues take place when it involves clustering unconditional data. Table 5.2 shows subspace clustering. The unconditional data are grouped into three classes.

**Table 5.2 Unconditional grouped class and descriptions**

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Natural Order</td>
<td>The need of an inherent natural order on the individual domains. This property renders an outsized variety of ancient similarity measures ineffective.</td>
</tr>
<tr>
<td>High Dimensionality</td>
<td>Unconditional datasets are commonly high dimensional</td>
</tr>
<tr>
<td>Subspace Clusters</td>
<td>Many unconditional datasets do not demonstrate clusters over the entire set of dimensions.</td>
</tr>
</tbody>
</table>

**Grid Density based Spatial Clustering**

Density based clustering approaches apply an area cluster criterion, during which clusters are considered regions within the data space during which the objects are dense and that are separated by regions of low object density. A general way to find areas of high density in the data space is based on grid cell densities. A histogram is created by dividing the data space into a number of non-overlapping regions.

Each region including a relatively large number of objects are possible cluster centers and the boundaries between clusters reduce in the valley of the histogram. The dimension of the cells establishes the calculations and the quality of the clustering. Cells of small dimensions will
give a very low estimate of the density, whereas large cells have a tendency to excessively smoothen the density estimate.

**CLIQUE Optimization in Spatial Clustering**

In spatial clustering, grid and density based approach, the size of the grid closely determines the computations and the class of the clustering. CLIQUE optimization, a density and grid based approach for high dimensional data sets, detects clusters in higher dimensional subspaces.

Density based approaches consider clusters as high density regions than their backgrounds. A general technique of finding high density regions in the data space is based on the grid cell densities.

The proposed CLIQUE optimization shows in Figure 5.8 takes the size of the grid and a global density threshold for clusters as input parameters. The computation density and the class of clustering are greatly dependent on these parameters.

An axis-parallelized histogram is constructed by partitioning the data space into a number of none hyper plane regions and then mapping the data points to each cell in the grid. Equal length intervals are used to partition each dimension, which results in uniform volume cells.

The density part of the subspace algorithm computes the clusters of the space. It has more common two input parameters viz. the length of the interval and the MinPts. In this study the MinPts parameter is used.
The initial step of the implementation process is selection of input. The spatial database and the MinPts clusters collected are put in the table and the diagram.

**Attribute Cleansing**

In this cleansing, the collected input data are classified and identified, as input data that are used for clustering and the data that are unnecessary for clustering. Attribute relative analysis is employed to reduce the unnecessary data before clustering. Every attribute is compared to MinPts. If the MinPts value is bigger than the attribute value, that attribute is not necessary for clustering.

**Build X. Y Parallel Histogram**

In this study using C++ coding the histogram is built in the map regions.
Rows are identified as X coordinates and Columns are identified as Y coordinates using the rightward-rule. Once the X, Y parallel histogram is created, it will split the database. 1. Reason for this study is to select the unit coordinates value initially 1. The Figure 5.9 shows the database splitting.

**Figure 5.9 The X, Y Axis Parallel Histogram for Database Splitting**

**Density Threshold Identifying**

```cpp
numberClusters = 0;
thresholdPoints = (int)(numberOfPoints * tau/100.0);
cout << "thresholdPoints = " << thresholdPoints << endl;
```

Comparing (dens region, threshold Point)

Threshold points have identified a number of points in every region for computing the density value of the region. The comparison operation identifies the density regions and the sparse regions.
Retrieve number of clusters and Optimization

for(i=1; i<cellrow; i++) { for(j="1;" j="")<cellcolumn; j++) { if(ptrcells[i][j].checked="" !=""1" & &"" ptrcells[i][j].clusterno="" 0"{ if(ptrcells[i][j].qualified="" >"" 0){ numberClusters++; if(retrieve(numberClusters, i, j)<0){ out << "retrieve error! \n"; return -1;} } else{ ptrcells[i][j].clusterNO =0; ptrCells[i][j].checked = 1; }

In this function, the study collects the number of clusters in the given space. Collect the X, Y coordinates are collected and put into pointer cells. The initial condition cluster no =0. The coordinates are moved into rightward rule to find the sparse region and the cluster region. If it reaches the cluster region the clusterNO is increased to 1, else clusterNO = 0.

Optimization is an important task of splitting the cluster region. In this study C++ user defined function is used to retrieve neighbor for checking the data space and dividing into more accurately.

int retrieve(int numberClusters, int i, int j)if( ptrCells[i][j].checked != 1 & & ptrCells[i][j].clusterNO == 0 ){ if( ptrCells[i][j].qualified > 0 ){ ptrCells[i][j].clusterNO = numberClusters; ptrCells[i][j].checked = 1; if( i != 0){ l = i-1; retrieve(numberClusters, l, j); } if( i != (cellRow-1) ) i=i+1; retrieve (numberClusters, l, j);} if( j != 0){ m = j-1; retrieve (numberclusters,i,m); } if( j != (cellColumn-1) ){m = j+1; retrieve (numberClusters, i, m); }

The first condition checks for whether any points have not been checked. If it ensures that, further execution will follow. Otherwise if checked
returns=1, there is no need for recursive optimization. Figure 5.10 shows the result of the retrieve function optimization separation.

Figure 5.10 Database Separation after Recursive Optimization

5.4 EXPERIMENTAL RESULTS

The dataset generator tool is used for analysis of the result. The Dataset generator computer program integrates several key parameters to generate strictly conjunctive classification rules which are then used to generate synthetic data sets. The present study uses the data generator tool for generating initial Chennai city crime datasets. Figure 5.11 shows the clustering results setting K= 15 crime density areas, MinPts= 30. Figure 5.11 shows splitting found by the CLIQUE Optimization algorithm on the spatial data set.
5.5 RESULT COMPARISON

Clique Optimization is fundamentally grid-density spatial clustering algorithm that uses to compare with a clique algorithm to accept the more excellent ability of new spatial clustering algorithm. Both the algorithms are run in the data generator tools that are used in this experiment.

The outcome of this part of the experiment is that the CLIQUE cannot find all natural clusters of spatial data space thoroughly as CLIQUE Optimization algorithm finds clusters on the whole. In this study, the cluster results of the CLIQUE and CLIQUE Optimization algorithm are compared as shown in Figure 5.12. The cluster result of the CLIQUE Optimization algorithm is more accurate and exact than the results of CLIQUE.
Figure 5.12 Comparison of CLIQUE and CLIQUE Optimization with Scalability and Data Points

5.6 CONCLUSION

High dimensional crime data are changing into increasingly common in police record fields. Because the range of dimensions increases, several clustering techniques begin to experience from the curse of dimensionality, demeaning the quality of the results. In high dimensional data, it becomes very sparse and distance measures become increasingly meaningless. This problem has been studied extensively and there are various solutions, each appropriate for different types of high dimensional data and data mining procedures.

CLIQUE Optimization algorithm performs the clustering and therefore the subspace choice simultaneously, however in small subspaces, adding one dimension at a time. This enables these algorithms to scale far more simply with each range quantity of instances within the dataset,
additionally because the number of attributes. However, performance drops quickly with the dimensions of the subspaces during which the clusters are found. The most essential parameter needed by these algorithms is that the density threshold. This will be difficult to learn, particularly across all dimensions of the dataset. Fortunately, although some dimensions are mistakenly ignored because of improper thresholds, the algorithms should realize the clusters during a smaller subspace.

Adaptive grid approaches facilitate to alleviate this drawback by permitting the amount of bins during a dimension to vary based on the characteristics of the data in that dimension. Often, CLIQUE Optimization algorithms are ready to realize clusters numerous shapes and sizes since the clusters are formed from various cells during a grid. This additionally means the clusters will overlap one another with one instance having the potential to be in additional than one cluster. It is additionally possible or an instance to be considered an outlier and does not belong to any cluster.