Chapter 2

Problem Definition and Related Work

In this chapter, we first define the subspace clustering problem, then we discuss the working of the CLIQUE [1] algorithm. A few improvements over the CLIQUE algorithm are also discussed.

2.1 Subspace Clustering Problem

The Subspace clustering problem was first introduced by R. Agrawal, in [1]. Subspace Clustering is the most informative/systematic approach for clustering high-dimensional data. It is the task of automatically identifying (in general several) subspaces of a high dimensional data space that allow better clustering of the data objects than the original data space [1].

Terminology Used:

Let $A = \{ A_1, A_2, \ldots, A_d \}$ be a set of bounded, totally ordered domains and $S = A_1 \times A_2 \times \ldots \times A_d$ a d-dimensional numerical space. $A_1, \ldots, A_d$ are referred to as the dimensions (attributes) of $S$.

The input consists of a set of d-dimensional points $V = \{ v_1, v_2, \ldots, v_m \}$ where $v_i = < v_{i1}, v_{i2}, \ldots, v_{id} >$. The jth component of $v_i$ is drawn from domain $A_j$. 
The data space $S$ is partitioned into non-overlapping rectangular units. The units are obtained by partitioning every dimension into $\xi$ intervals of equal length, which is an input parameter.

Each unit $u$ is the intersection of one interval from each attribute. It has the form $\{u_1, \ldots, u_d\}$ where $u_i = [l_i, h_i)$ is a right-open interval in the partitioning of $A_i$.

A point $v = \{v_1, v_2, \ldots, v_d\}$ is contained in a unit $u = \{u_1, u_2, \ldots, u_d\}$ if $l_i \leq v_i < h_i$ for all $u_i$.

The selectivity of a unit is defined to be the fraction of the total data points contained in the unit. A unit $u$ is called a dense unit if $\text{selectivity}(u)$ is greater than $\tau$, the density threshold which is input by the user.

A $k$-dimensional subspace is a projection of the data set $V$ into $A_{i_1} \times A_{i_2} \times \ldots \times A_{i_k}$, where $k < d$ and $t_i < t_j$ if $i < j$. A $k$-dimensional unit $u^k$ in this subspace is the intersection of an interval from each of the $k$ attributes.

A cluster is a maximal set of connected dense units in $k$-dimensions. Two $k$-dimensional units $u_1, u_2$ are connected if they have a common face or if there exists
another k-dimensional unit \( u_3 \) such that \( u_1 \) is connected to \( u_3 \) and \( u_2 \) is connected to \( u_3 \).

Units \( u_1^k = \{ r_{i1}, \ldots, r_{ik} \} \) and \( u_2^k = \{ r'_{i1}, \ldots, r'_{ik} \} \) have a common face if there are \( k-1 \) dimensions, assume dimensions \( A_{i1}, \ldots, A_{ik-1} \), such that \( r_{ij} = r'_{ij} \) for \( j = 1 \) to \( k-1 \) and either \( h_{ik} = l'_{ik} \) or \( h'_{ik} = l_{ik} \).

A region in \( k \) dimensions is an axis-parallel rectangular \( k \)-dimensional set. Regions are considered as unions of units. Region \( R \) is said to be contained in a cluster \( C \) if \( R \cap C = R \).

A region \( R \) contained in a cluster \( C \) is said to be maximal if no proper superset of \( R \) is contained in \( C \).

A minimal description of a cluster is a non-redundant covering of the cluster with maximal regions. That is, a minimal description of a cluster \( C \) is a set \( R \) of maximal regions such that their union equals \( C \) but the union of any proper subset of \( R \) does not equal \( C \).

The Problem: Given a set of data points and the input parameters \( \xi \) and \( \tau \), find clusters in all subspaces of the original data space and present a minimal description of each cluster in the form of a DNF expression.
CHAPTER 2. PROBLEM DEFINITION AND RELATED WORK

Figure 2.1: Illustration of CLIQUE definitions.

Example 2.1:

In Figure 2.1, the two dimensional space (age, salary) has been partitioned by a 10 X 10 grid.

A unit is the intersection of intervals; above an example of a 2-dimensional unit \( u = (30 \leq \text{age} < 35) \land (1 \leq \text{salary} < 2) \).

A region is a rectangular union of units. A and B are both regions: \( A = (30 \leq \text{age} < 50) \land (4 \leq \text{salary} < 8) \) and \( B = (40 \leq \text{age} < 60) \land (2 \leq \text{salary} < 6) \).

The minimal description for the cluster \( (A \cup B) \) is the DNF expression:

\[
A = (30 \leq \text{age} < 50) \land (4 \leq \text{salary} < 8) \lor (40 \leq \text{age} < 60) \land (2 \leq \text{salary} < 6).
\]
2.2 CLIQUE Algorithm

The CLIQUE algorithm consists of the following three steps:

1. Identification of subspaces that contain clusters.
2. Identification of clusters.
3. Generation of minimal description for the clusters.

The main part of step 1 consists of finding the dense units in different subspaces. The dense units are identified using a bottom-up algorithm that exploits the monotonicity of the clustering criterion with respect to dimensionality to prune the search space. This algorithm is similar to the apriori algorithm for mining association rules [4].

Example 2.2 Let the transaction database, TDB, be Table 2.1 consisting of a total of 10 transactions, with 6 numeric attributes each and the user input values of $\xi$ and $\tau$ be 5 and 0.2 respectively. The missing values for the attributes are represented by a '?' symbol. Assume for the sake of simplicity that all the attribute values range from 1 to 100.
**CHAPTER 2. PROBLEM DEFINITION AND RELATED WORK**

<table>
<thead>
<tr>
<th>Tid</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1</td>
<td>21</td>
<td>41</td>
<td>?</td>
<td>4</td>
<td>18</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
<td>24</td>
<td>44</td>
<td>9</td>
<td>9</td>
<td>49</td>
</tr>
<tr>
<td>30</td>
<td>6</td>
<td>26</td>
<td>46</td>
<td>45</td>
<td>23</td>
<td>83</td>
</tr>
<tr>
<td>40</td>
<td>9</td>
<td>29</td>
<td>?</td>
<td>57</td>
<td>?</td>
<td>5</td>
</tr>
<tr>
<td>50</td>
<td>2</td>
<td>8</td>
<td>25</td>
<td>58</td>
<td>78</td>
<td>30</td>
</tr>
<tr>
<td>60</td>
<td>53</td>
<td>?</td>
<td>92</td>
<td>59</td>
<td>?</td>
<td>52</td>
</tr>
<tr>
<td>70</td>
<td>19</td>
<td>8</td>
<td>89</td>
<td>58</td>
<td>78</td>
<td>57</td>
</tr>
<tr>
<td>80</td>
<td>82</td>
<td>2</td>
<td>?</td>
<td>52</td>
<td>72</td>
<td>12</td>
</tr>
<tr>
<td>90</td>
<td>89</td>
<td>78</td>
<td>10</td>
<td>25</td>
<td>?</td>
<td>38</td>
</tr>
<tr>
<td>100</td>
<td>?</td>
<td>68</td>
<td>75</td>
<td>?</td>
<td>62</td>
<td>13</td>
</tr>
</tbody>
</table>

Table 2.1: A transaction database TDB.

*CLIQUE* finds the dense units for identification of the subspaces containing clusters as follows –

1. Each attribute is split into $\xi$ intervals to form $\xi$ 1-dimensional candidate units for each attribute namely $A_1, \ldots, A_5, B_1, \ldots, B_5, C_1, \ldots, C_5, D_1, \ldots, D_5$ and so on till $F_1, \ldots, F_5$.

Hence in this example we will have a total of $5 \times 6 = 30$ 1-dimensional candidate units.
2. By doing a first pass over the above dataset, the frequency count of all these 1-dimensional candidate units is found. Selectivity of a unit is equal to the frequency count of that unit divided by the total number of transactions. Those units whose selectivity is greater than 0.2 are identified as 1-dimensional dense units $D_1$. $D_1$ in this example is \{A1, A9, B1, B3, C5, D6, E1, E8, F2, F6\}

3. The 2-dimensional candidate units $C_2$, are generated by forming all possible pairs of the 1-dimensional dense units $D_1$. Some candidate units are pruned. Only those candidate units are retained which have all its subset units dense. A 2-dimensional candidate unit $u_1^i u_1^j \in C_2$ if and only if $u_1^i, u_1^j \in D_1$. In this example, $C_2$ consists of \{A1B1, A1B3, ..., E8F2, E8F6\}

4. A second pass is made through the dataset to find the selectivity of all the two dimensional candidate units $u_2^i \in C_2$ for $i = 1$ to $n$, $n$ representing the total number of 2-dimensional candidate units. Thus we get, $D_2$ consisting of \{A1B3, B3C5\}

5. For $k \geq 3$, the candidate units generation procedure and procedure used for pruning the generated candidate units is as given below -

The candidate generation procedure used for generating $C_k$ from $D_{k-1}$ is as under –

\[
\text{insert into } C_k \\
\text{select } u_1[l_1, h_1), u_1[l_2, h_2), ..., u_1[l_{k-1}, h_{k-1}), u_2[l_{k-1}, h_{k-1}) \\
\text{from } D_{k-1} u_1, D_{k-1} u_2 \\
\text{where } u_1.a_1 = u_2.a_1, u_1.l_1 = u_2.l_1, u_1.h_1 = u_2.h_1,
\]
\[ u_{1,2} = u_{2,2}, \ u_{1,2} = u_{2,2}, \ u_{1,2} = u_{2,2}, \ldots, \]
\[ u_{1, k-2} = u_{2, k-2}, \ u_{1, k-2} = u_{2, k-2}, \ u_{1, k-2} = u_{2, k-2}, \]
\[ u_{1, k-1} < u_{2, k-1} \]

In the above pseudo-code for the join operation, \( u_{a, i} \) represents the \( i \)th dimension or attribute of unit \( u \) and \( u_{[h_i, h_i]} \), represents its interval in the \( i \)th dimension.

Pruning procedure used for \( k \)-dimensional candidate units \( C_k \) - All those \( C_k \) units which do not have all its \((k-1)\) dimensional subsets in the set of \((k-1)\) dimensional dense units are discarded from the set of \( C_k \) units generated above.

Then the \( k^{th} \) pass is done to find selectivity of all \( C_k \) units and obtain the \( D_k \) units.

This process is continued till no candidate units can be derived or no candidate is dense. In this manner all the dense units belonging to the different subspaces are found. These units form the input for the second step of \( CLIQUE \).

**Time complexity:** If \( k \) is the highest dimensionality of any dense unit and \( m \) is the number of the input points, the above algorithm will make \( k \) passes over the database. If a dense unit exists in \( k \) dimensions, then all of its projections in a subset of the \( k \) dimensions that is, \( O(2^k) \) different combinations will also be dense. Hence, the time complexity of this algorithm is \( O(c^k + mk) \) for a constant \( c \).

The second step of \( CLIQUE \) takes as input the set of dense units \( D \), all in the same \( k \)-dimensional space \( S \) and outputs a partition of \( D \) into \( P_1, \ldots, P_q \), such that all
units in $P_i$ are connected and no two units $u_i \in P_i$, $u_j \in P_j$ with $i \neq j$ are connected. All these partitions represent the clusters found in the k-dimensional space $S$. It finds the partitions by using a depth-first search algorithm to find the connected components in the graph formed by representing the dense units as the vertices of the graph. An edge exists between those vertices whose corresponding dense units have a common face.

The step three takes as input the clusters identified in step two and generates a concise description for it. For this purpose it first uses a greedy growth method to cover the clusters by a number of maximal rectangles(regions), and then discards the redundant rectangles to generate a minimal cover.

Some of the drawbacks of the CLIQUE algorithm are as under-

- It does not provide any support to the user for selecting the values for the input parameters $\xi$ and $\tau$. The cluster boundaries generated are totally dependant on the value of $\xi$ and the value of $\tau$ decides the quality of the clusters that will be generated. If the value of $\tau$ is set too low then we will get a large number of dense units, and some of the clusters that we get from these dense units will be redundant. Similarly, if the value is too high then we will miss to capture some significant clusters.
• It is tedious to make repeated passes over the database to find the selectivity of the large number of candidate units generated. This condition worsens when the dimensionality of the subspace clusters found in the database increases. As the dimensionality of the subspace clusters increases, there is an explosion in the number of dense and the candidate units generated. CLIQUE uses a MDL-based pruning technique. In this the dense units in the subspaces with low coverage are pruned so as to reduce the number of dense and candidate units generated. The coverage of a subspace is the fraction of the database that is covered by the dense units. This is believed to make the algorithm faster but it may lead to missing out of some important clusters.

• If the size of the dataset is very large both with respect to the number of records and the number of attributes (data dimensionality), the time taken for each database pass to find the selectivity of the candidate units will increase substantially.

2.3 Improvements over CLIQUE

In the past few years, some subspace clustering algorithms have been proposed to overcome some of the problems of the CLIQUE algorithm.
The ENCLUS \cite{8}, a ENtropy-based subspace CLUstering algorithm was proposed to handle the large number of subspaces with clusters within them. In CLIQUE, the MDL-based pruning technique was used to prune some subspaces with low coverage to make the algorithm faster. However, it had the trade-off of missing out some significant dense units found in subspaces with low coverage. The ENCLUS \cite{8} algorithm has made the following contributions to the subspace clustering problem –

- It has identified the following additional criteria for determining subspaces with good clustering:
  a) Criterion of High Coverage
  b) Criterion of high density and
  c) Correlation of dimensions

![Figure 2.2: Examples of two data sets with equal coverage but different densities. The area within the rectangles is the value of the coverage.](image)

In the figure 2.2 cases (a) and (b) have the same coverage, however the points in (a) are more closely packed and is a better candidate to qualify as a cluster.

- **ENCLUS [8]** uses the entropy metric to measure all the above three criteria simultaneously to find subspaces with good clustering. It is also a grid based method and takes the inputs for the threshold from the user. In order to calculate the entropy, it also divides each dimension into equal width intervals to form a grid. Hence the size selected for the intervals, affects the quality of the final clustering obtained.

- **MAFIA [16]** (Merging of Adaptive Finite Intervals) is another subspace clustering algorithm, which uses adaptive interval sizes to partition the dimension depending on the distribution of data in the dimension. Using adaptive grid sizes, **MAFIA** attempted to reduce the computation and improve the clustering quality by concentrating on the portion of the data space which have more data points and thus more likelihood of having clusters.

- **PROCLUS [2]** uses the concept of PROjected CLUStering for finding clusters in a multi-dimensional data space. **PROCLUS** also discovers interesting correlations among the data in various subspaces of the original high dimensional space, but it differs from **CLIQUE** in the output produced. It outputs a partition of the data points into clusters, together with the sets of
dimensions on which points in each cluster are correlated. The clusters output by PROCLUS are useful in applications like classification and trend analysis where it is required to partition the data points into disjoint partitions. It fails to detect any overlapping clusters existing in the data set. ORCLUS [3] is also an example of a projected clustering algorithm.