Chapter 1

Introduction

1.1 Background

The amount of raw data and information being captured and stored in computer files and databases in almost every field has been growing at a tremendous pace. In short we can say that we have been flooded with data but we are still starving to get the knowledge from this vast pool of existing data. In today's competitive world, all concerned need to extract as much information as possible from their data sources to help in efficient decision making, so as to compete with their rivals and achieve their goals. Data mining comes into play to help users satisfy such needs. Data mining, which is also referred to as knowledge discovery in databases, means a process of nontrivial extraction of implicit, previously unknown and potentially useful information (such as knowledge rules, constraints, regularities) from data in databases [26]. Data mining combines methods and tools from at least three areas namely machine learning, statistics, and databases [19].

Clustering is a data mining technique that helps in identifying clusters within the domain space and has many applications in several fields. As a data mining task, data clustering also referred to as unsupervised classification can be thought of as partitioning or segmenting the data into groups that might or might not be disjoint. Data clustering has been studied in statistics [10,20], machine learning [14,15], and spatial
data mining [10, 11, 25] areas with different emphasis. The unsupervised nature of clustering makes it applicable to applications, where the user has limited domain knowledge. Some of the current applications, which use clustering techniques extensively are clustering of web-search results and clustering of spatial databases. Most of the traditional clustering algorithms have been designed to discover clusters in the full dimensional space using various distance functions.

1.2 Motivation

In recent years, there has been an increase in the number of new database applications dealing with very large high dimensional data sets. These applications to name a few include multimedia content-based retrieval, geographic and molecular biology data analysis, text mining, bio-informatics, medical applications, and time-series matching. These applications place special requirements on clustering algorithms: the ability to find good quality clusters embedded in subspaces of high dimensional data preferably without taking any inputs from the user (which requires the user to have good domain knowledge), scalability, end-user comprehensibility of the results, non-presumption of any canonical data distribution, and insensitivity to the order of input records. Clustering algorithms which work on the full dimensional space of the data fail to find clusters in high dimensional datasets due to the following main reasons – the average density of points anywhere in the high dimensional data space is likely to be
low [6]. Secondly, in the high dimensional data there are more chances of having missing values in the data attributes. In order to apply the full dimensional clustering algorithms, these missing values are normally replaced by values taken from a random distribution say X. Here an assumption is made that, the attribute containing missing values, follows that particular X distribution. This assumption need not be true always and thereby affect the quality of the clustering results obtained. Majority of the traditional clustering algorithms are sensitive to the order of input records and require input parameters from the user.

The subspace clustering algorithm CLIQUE [1] satisfies some of the above requirements. It identifies the subspace clusters in the high dimensional data by finding all the sets of connected dense units existing in the various subspaces. It presents the cluster descriptions in the form of DNF expressions that are minimized for easy interpretation. It produces identical results irrespective of the order of the input records and does not need to make any assumptions about the data distributions for any attributes to handle any missing values. However, it requires the user to give the inputs, \( \tau \) (threshold value) and \( \xi \) (number of intervals) in order to find the dense units. Hence the accuracy of the results obtained depends on the values input by the user. It uses the level-wise \textit{apriori} [4] algorithm for finding the dense units. Hence suffers from the same problems as the \textit{apriori} algorithm in the following situations:

- If the user inputs a large value for \( \xi \) or enters a very low value for \( \tau \), the number of candidate and dense units generated will be huge in number. And as a result
the first step of CLIQUE to identify the dense units in the different subspaces will be computationally very expensive.

- If the dimensionality of the clusters is large, then the database will have to be scanned a large number of times to find the high dimensional dense units. And, if the size of the database is also very large then it will still add to the time complexity.

As a result of the emerging real life data applications, there is a demand for clustering algorithms, which can efficiently identify good quality clusters from huge, high dimensional data sets. Hence, developing efficient techniques to find clusters in huge, high dimensional data sets has become an important research direction in data mining.

1.3 Contributions

In this thesis, we study the problem of subspace clustering for very high dimensional huge data sets with missing values. In particular, we make the following contributions -

- **Efficient storage structure**: Based on the properties of very high dimensional huge data sets containing missing values, and the requirements of the subspace clustering
algorithms, we have developed an Attribute Oriented Storage Structure (AOSS) for storing very high dimensional huge data sets.

- **Scalability:** With the increasing size of the databases, we need to have subspace clustering algorithms, which can be used for very large data sets. We have used the sampling technique to address this issue. The *SAMCLIQ* algorithm developed using sampling technique gave us very efficient results when compared with the *CLIQUE* algorithm.

- **Efficiency:** To handle this issue, we have used a depth-first approach and the concept of maximal dense units for identifying the subspaces containing the clusters. The subspace clustering algorithms CLIQUE [1] and MAFIA [16] have used the level-wise *apriori* algorithm for identifying the dense units. Again here we used the AOSS method of storage representation and found that it gives very good results for very high dimensional huge datasets with missing value attributes.

- **Applicability:** We extended the AOSS method using the maximal dense unit concept to find clusters in datasets containing attributes with varied threshold requirements. As an application of this technique in applications like census data analysis, we developed a subspace clustering algorithm to allow mining of all the subspace clusters found in the clusters identified in the original dataset.
1.4 Organization of the Thesis

The remainder of the thesis is structured as follows:

- In Chapter 2, we present the subspace clustering problem and an overview of the related work carried out in high-dimensional clustering.

- In chapter 3, An Attribute Oriented Storage Structure (AOSS) for storing very high dimensional datasets with many missing values has been developed. The reduction in time complexity using this structure is reported along with the experimental results obtained using synthetic datasets.

- In Chapter 4, a sampling based subspace clustering algorithm \textit{SAMCLIQ} [36] is developed to handle very large data sets. The experimental evaluation and performance study by comparing with CLIQUE has been carried out.

- In Chapter 5, details of algorithms developed using AOSS based structure for finding maximal dense units with uniform threshold value (MADUGEN) and multiple threshold values (MADUGENMT) have been discussed. Using AOSS structure and MADUGENMT a subspace clustering algorithm AOMLSCLUS, has been presented and its application for analyzing census data discussed.
• In Chapter 6, we summarize the characteristics of the AOSS method along with a discussion of some interesting extensions and applications of subspace clustering using AOSS.

• In Chapter 7, we conclude with a few directions for future work.