Evaluation of Subspace Clustering of High Dimensional Data

V.Kumutha, Asst. Prof.
Department of Computer Science
D.J. Academy for Managerial Excellance
Coimbatore, Tamilnadu, India
kumuthav@gmail.com

S.Palaniammal, Prof. & Head
Dept. of Science & Humanities
V.L.B.Janakiammal College of Engg. & Tech.
Coimbatore, Tamilnadu, India
splvlb@yahoo.com

Abstract— Data Clustering is an unsupervised method for extracting hidden pattern from huge datasets. Due to the sparsity of data points, conventional clustering algorithms do not scale well to cluster high dimensional data sets in terms of effectiveness and efficiency. The curse of dimensionality is the challenge posed by high dimensional data. Subspace clustering or projected clustering are the solutions for finding clusters and their relevant attributes from a dataset. In this paper, we evaluate various subspace clustering algorithms. A comparative empirical evaluation is done by using the data sets taken from UCI ML Repository.

Keywords—clustering, high dimensional, subspace clustering

I. INTRODUCTION

Data mining is the process of extracting potentially useful information from a data set [1]. Clustering is a popular data mining technique which is intended to help the user discover and understand the structure or grouping of the data in the set according to a certain similarity measure [2]. Clustering algorithms usually employ a distance metric (e.g., Euclidean) or a similarity measure in order to partition the database so that the data points in each partition are more similar than points in different partitions.

High dimensionality poses two challenges for unsupervised learning algorithms. First, the presence of irrelevant and noisy features can mislead the clustering algorithm [3]. Second, in high dimensions, data may be sparse (the curse of dimensionality), making it difficult for an algorithm to find any structure in the data. To ameliorate these problems, two basic approaches for reducing the dimensionality have been investigated: feature subset selection and feature transformations, which project high dimensional data onto “interesting” subspaces [4][8]. If different subsets of the points cluster well on different subspaces of the feature space, a global dimensionality reduction will fail. Subspace clustering is an extension of traditional clustering algorithm which considers all of the dimensions of an input dataset in an attempt to learn as much as possible about each object described. In high dimensional data, however, many of the dimensions are often irrelevant. These irrelevant dimensions confuse clustering algorithms by hiding clusters in noisy data. In very high dimensions it is common for all of the objects in a dataset to be nearly equidistant from each other, completely masking the clusters.

Feature selection methods have been used somewhat successfully to improve cluster quality. These algorithms find a subset of dimensions on which to perform clustering by removing irrelevant and redundant dimensions. The problem with feature selection arises when the clusters in the dataset exist in multiple, possibly overlapping subspaces [2]. Subspace clustering algorithms attempt to find such clusters.

Different approaches to clustering often define clusters in different ways [2]. One type of clustering creates discrete partitions of the dataset, putting each instance to one group. Some, like k-means, put every instance into one of the clusters. Others allow for outliers, which are defined as points that do not belong to any of the clusters. Another approach to clustering creates overlapping clusters, allowing an instance to belong to more than one group. Usually these methods also allow an instance to be an outlier, belonging to no particular cluster. No one type of clustering is better than the others, but some are more appropriate to certain problems. Domain specific knowledge about the data is often very helpful in determining which type of cluster formation will be the most useful.

In subspace clustering, each cluster is mapped to an associated subspace, which allows for more flexibility than global methods projecting the entire data set onto a single subspace [6]. In general, these algorithms suffer from two common limitations, have problems with subspace clusters of significantly different dimensionality and they often fail to discover clusters of different shape and densities.

Subspace clustering aims at computing all clusters in all subspaces of the feature space. The information of objects clustered differently in varying subspaces is conserved. Objects may be assigned to several clusters (in different subspaces). In this paper, we evaluate various subspace clustering methods such as DOC, Fires, INSCY, SCHISM and SUBCLU.

This paper is organized as follows: Section 2 covers the literature review of the algorithms. The empirical comparison of various algorithms was done in Section 3. Conclusion is given in Section 4.
A. Density-based Clustering

Agrawal et al. proposed CLIQUE algorithm [5], which was one of the first subspace clustering algorithms. The algorithm combines density and grid based clustering and uses an APRIORI style search technique to find dense subspaces. Once the dense subspaces are found they are sorted by coverage, defined as the fraction of the dataset the dense units in the subspace represent. The subspaces with the greatest coverage are kept and the rest are pruned. The algorithm then finds adjacent dense grid units in each of the selected subspaces using a depth first search. Clusters are formed by combining these units using a greedy growth scheme. The algorithm starts with an arbitrary dense unit and greedily grows a maximal region in each dimension until the union of all the regions covers the entire cluster. Redundant regions are removed by a repeated procedure where smallest redundant regions are discarded until no further maximal region can be removed. The hyper-rectangular clusters are then defined by a Disjunctive Normal Form (DNF) expression.

The region growing, density based approach for generating clusters allows CLIQUE to find clusters of arbitrary shape, in any number of dimensions. Clusters may be found in the same, overlapping, or disjoint subspaces. The DNF [7] expressions used to represent clusters are often very interpretable and can describe overlapping clusters, meaning that instances can belong to more than one cluster. This is often advantageous in subspace clustering since the clusters often exist in different subspaces and thus represent different relationships.

MAFIA[10] is another extension of CLIQUE that uses an adaptive grid based on the distribution of a data to improve efficiency and cluster quality. MAFIA also introduces parallelism to improve scalability. MAFIA initially creates a histogram to determine the minimum number of bins for a dimension. The algorithm then combines adjacent cells of similar density to form larger cells. The dimension is partitioned based on the data distribution and the resulting boundaries of the cells capture the cluster perimeter more accurately than fixed sized grid cells. Once the bins have been defined, MAFIA proceeds much like CLIQUE, using an APRIORI style algorithm to generate the list of clusterable subspaces by building up from one dimension. It finds any number of clusters of arbitrary shape in subspaces of varying size. It also represents clusters as DNF expressions. MAFIA performs many times faster than CLIQUE on similar datasets.

B. Projected Clustering

PROCLUS PROjected CLUStering [4] was the first top-down subspace clustering algorithm. Similar to CLARANS, PROCLUS samples the data, then selects a set of k-medoids and iteratively improves the clustering. The algorithm uses a three phase approach consisting of initialization, iteration, and cluster refinement. Initialization selects a set of potential medoids that are far apart using a greedy algorithm. The iteration phase selects a random set of k medoids from this reduced dataset, replaces bad medoids with randomly chosen new medoids, and determines if clustering has improved. Cluster quality is based on the average distance between instances and the nearest medoid. For each medoid, a set of dimensions is chosen whose average distances are small compared to statistical expectation. Once the subspaces have been selected for each medoid, average Manhattan segmental distance is used to assign points to medoids, forming clusters. The refinement phase computes a new list of relevant dimensions for each medoid based on the clusters formed and reassigns points to medoids, removing outliers.

The distance based approach of PROCLUS is biased toward clusters that are hyper-spherical in shape. Also, while clusters may be found in different subspaces, the subspaces must be of similar sizes since the user must input the average number of dimensions for the clusters. Clusters are represented as sets of instances with associated medoids and subspaces and form non-overlapping partitions of the dataset with possible outliers. PROCLUS is actually somewhat faster than CLIQUE due to the sampling of large datasets. However, using a small number of representative points can cause PROCLUS to miss some clusters entirely.

ORCLUS Arbitrarily ORiented projected CLUSter generation [9] is an extended version of the algorithm PROCLUS [4] that looks for non-axes parallel subspaces. This algorithm arose from the observation that many datasets contain inter-attribute correlations. The algorithm can be divided into three steps: assign clusters, subspace determination, and merge. During the assign phase, the algorithm iteratively assigns data points to the nearest cluster centers. The distance between two points is defined in a subspace E, where E is a set of orthonormal vectors.
in some d-dimensional space. Subspace determination redefines the subspace E associated with each cluster by calculating the covariance matrix for a cluster and selecting the orthonormal eigenvectors with the least spread (smallest eigen values). Clusters that are near each other and have similar directions of least spread are merged during the merge phase. The number of clusters and the size of the subspace dimensionality must be specified. The authors provide a general scheme for selecting a suitable value. A statistical measure called the cluster sparsity coefficient, is provided which can be inspected after clustering to evaluate the choice of subspace dimensionality.

DOC [11] is a Monte Carlo algorithm that blends the grid based approach used by the bottom-up approaches and the iterative improvement method from the top-down approaches. A projective cluster is defined as a pair (C;D) where C is a subset of the instances and D is a subset of the dimensions of the dataset. The goal is to find a pair where C exhibits a strong clustering tendency in D. To find these optimal pairs, the algorithm creates a small subset X, called the discriminating set, by random sampling. This set can is used to differentiate between relevant and irrelevant dimensions for a cluster. For a given a cluster pair (C,D), instances p in C, and instances q in X the following should hold true: for each dimension i in D, |q(i) - p(i)| <= w, where w is the fixed side length of a subspace cluster, given by the user. p and X are both obtained through random sampling and the algorithm is repeated with the best result being reported.

SUBCLU is able to detect arbitrarily shaped and positioned shaped clusters in subspaces. It improves the cluster definition by variable thresholds adapting to the subspace dimensionality. Using a monotonicity property, SUBCLU reduces the search space by pruning higher dimensional projections like CLIQUE. In contrast to grid-based approaches, the density-based paradigm uses the original data and requires expensive database scans for each ε-neighborhood computation. This results in an inefficient computation. A more efficient, however, approximative solution is proposed by FIRES[12]. Instead of going through the subspaces bottom up, FIRES uses 1d histogram information to jump directly to interesting subspace regions.

### III. EXPERIMENTAL EVALUATION

#### A. Data set

4 data sets were taken for experimental comparison of subspace clustering algorithms. All the data sets were taken from UCI repository. Table 1 shows the details of the data sets taken for comparison.

#### B. Empirical Comparison

In this section, we compare the algorithms based on the performance of searching techniques and evaluation criteria. We also examined how well each of the algorithms was able to determine the correct subspaces for each cluster. The performance result of each of the algorithm is shown in the following table. The algorithm chosen for the comparison were SUBCLU, Fires, INSCY, SCHISM and DOC.

Figure 2 shows the comparison of accuracy, coverage and entropy on breast cancer, liver and diabetes data set for subspace clustering methods.

#### C. Result Analysis

The subspace clustering methods were implemented using Weka 3.6.1. The algorithms SCHISM and SUBCLU produced an accuracy between .65 and .70 for breast cancer and diabetes dataset. The algorithms SCHISM and SUBCLU produced a good coverage of 1 which determines the quality of clusters.

The clusters accuracy was compared and also determines the extent to which subspace clusters successfully generalizes the underlying data distribution. Entropy determines the purity of clustering. Coverage shows the size of cluster.

#### Table 1: Data sets – Number of Instances and Features

<table>
<thead>
<tr>
<th>Data set</th>
<th>Number of Instances</th>
<th>Number of Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breast Cancer</td>
<td>286</td>
<td>10</td>
</tr>
<tr>
<td>Liver</td>
<td>345</td>
<td>7</td>
</tr>
<tr>
<td>Diabetes</td>
<td>768</td>
<td>9</td>
</tr>
</tbody>
</table>

#### Table 2: Comparison of subspace and projected clustering methods on 3 real datasets

<table>
<thead>
<tr>
<th>Data set</th>
<th>Algorithm</th>
<th>No. of clusters</th>
<th>Accuracy</th>
<th>Coverage</th>
<th>Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breast Cancer</td>
<td>DOC</td>
<td>2</td>
<td>.7</td>
<td>1</td>
<td>.13</td>
</tr>
<tr>
<td></td>
<td>Fires</td>
<td>1</td>
<td>.7</td>
<td>.82</td>
<td>.09</td>
</tr>
<tr>
<td></td>
<td>INSCY</td>
<td>1</td>
<td>.7</td>
<td>.71</td>
<td>.15</td>
</tr>
<tr>
<td></td>
<td>SCHISM</td>
<td>2757</td>
<td>.69</td>
<td>1</td>
<td>.31</td>
</tr>
<tr>
<td></td>
<td>SUBCLU</td>
<td>511</td>
<td>.7</td>
<td>1</td>
<td>.13</td>
</tr>
<tr>
<td>Liver</td>
<td>DOC</td>
<td>2</td>
<td>.58</td>
<td>.98</td>
<td>.03</td>
</tr>
<tr>
<td></td>
<td>Fires</td>
<td>5</td>
<td>.57</td>
<td>.14</td>
<td>.02</td>
</tr>
<tr>
<td></td>
<td>INSCY</td>
<td>1</td>
<td>.59</td>
<td>.11</td>
<td>.01</td>
</tr>
<tr>
<td></td>
<td>SCHISM</td>
<td>2076</td>
<td>.63</td>
<td>1</td>
<td>.36</td>
</tr>
<tr>
<td></td>
<td>SUBCLU</td>
<td>334</td>
<td>.59</td>
<td>1</td>
<td>.06</td>
</tr>
<tr>
<td>Diabetes</td>
<td>DOC</td>
<td>2</td>
<td>.65</td>
<td>.98</td>
<td>.07</td>
</tr>
<tr>
<td></td>
<td>Fires</td>
<td>7</td>
<td>.65</td>
<td>.24</td>
<td>.35</td>
</tr>
<tr>
<td></td>
<td>INSCY</td>
<td>4</td>
<td>.65</td>
<td>.08</td>
<td>.3</td>
</tr>
<tr>
<td></td>
<td>SCHISM</td>
<td>5454</td>
<td>.65</td>
<td>1</td>
<td>.34</td>
</tr>
<tr>
<td></td>
<td>SUBCLU</td>
<td>2661</td>
<td>.7</td>
<td>1</td>
<td>.2</td>
</tr>
</tbody>
</table>
IV. CONCLUSION

High dimensional data is increasingly common in many fields. As the number of dimensions increase, many clustering techniques begin to suffer from the curse of dimensionality, degrading the quality of the results. In high dimensions, data 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.

Subspace clustering attempts to integrate feature evaluation and clustering in order to find clusters in different subspaces. In this paper, we have shown that the SUBCLU, PROCLUS, Fires, INSCY, SCHISM and DOC clustering algorithms are implemented and compared based on accuracy and coverage. From the result we have shown that SCHISM and SUBCLU gives better accuracy and coverage for breast cancer and diabetes dataset.

REFERENCES