Available online at http://www.ijcrls.com

International Journal of Current Research in Life Sciences
Vol. 07, No. 01, pp.829-836, January, 2018

ISSN: 2319-9490

RESEARCH ARTICLE

HIGH DIMENSIONAL DATA CLUSTERING

[1,*]Pavithra, M. and [2]Dr. Parvathi, R.M.S.

[1]Assistant Professor, Department C.S.E, Jansons Institute of Technology, Coimbatore, India
[2]Dean- PG Studies, Sri Ramakrishna Institute of Technology, Coimbatore, India

Received 14th November, 2017; Accepted 09th December, 2017; Published Online 30th January, 2018

ABSTRACT

Clustering becomes difficult due to the increasing sparsity of such data, as well as the increasing difficulty in distinguishing distances between data points. The proposed method called "kernel trick" and "Collective Neighbour Clustering", which takes as input measures of correspondence between pairs of data points. Real-valued hubs are exchanged between data points until a high-quality set of patterns and corresponding clusters gradually emerges (Aggarwal *et al*., 2015). To validate our theory by demonstrating that hubness is a high-quality measure of point centrality within a high dimensional information cluster, and by proposing several hubness-based clustering algorithms, showing that main hubs can be used effectively as cluster prototypes or as guides during the search for centroid-based cluster patterns (Gnanabaskaran *et al*., 2011). Experimental results demonstrate the good performance of our proposed algorithms in manifold settings, mainly focused on large quantities of overlapping noise. The proposed methods are modified mostly for detecting approximately hyper spherical clusters and need to be extended to properly handle clusters of arbitrary shapes (Naveen *et al*., 2011). For this purpose, we provide an overview of approaches that use quality metrics in high-dimensional data visualization and propose systematization based on a thorough literature review. We carefully analyze the papers and derive a set of factors for discriminating the quality metrics, visualization techniques, and the process itself (David, 2010). The process is described through a reworked version of the well-known information visualization pipeline. We demonstrate the usefulness of our model by applying it to several existing approaches that use quality metrics, and we provide reflections on implications of our model for future research. High-dimensional data arise naturally in many domains, and have regularly presented a great challenge for traditional data-mining techniques, both in terms of effectiveness and efficiency (Sembiring *et al*., 2010). Clustering becomes difficult due to the increasing sparsity of such data, as well as the increasing difficulty in distinguishing distances between data points. In this paper we take a novel perspective on the problem of clustering high-dimensional data (Singh, 2012). Instead of attempting to avoid the curse of dimensionality by observing a lower-dimensional feature subspace, we embrace dimensionality by taking advantage of some inherently high-dimensional phenomena. More specifically, we show that hubness, i.e., the tendency of high-dimensional data to contain points (hubs) that frequently occur in k-nearest neighbour lists of other points, can be successfully exploited in clustering. We validate our hypothesis by proposing several hubness-based clustering algorithms and testing them on high-dimensional data. Experimental results demonstrate good performance of our algorithms in multiple settings, particularly in the presence of large quantities of noise (Mohamed *et al*., 2009).

Key words: Data Mining, Clustering, Semi Supervised Clustering, High Dimensional Data, Clique, Enclus, Mafia, O-Cluster, Optigrid.

Citation: ***Pavithra, M. and Dr.Parvathi, R.M.S.*, 2018.** "High Dimensional Data Clustering" *International Journal of Current Research in Life Sciences*, 7, (01), 829-836.

INTRODUCTION

Clustering in general is an unsupervised process of grouping elements together, so that elements assigned to the same cluster are more similar to each other than to the remaining data points (Aggarwal, 2014). This goal is often difficult to achieve in practice. Over the years, various clustering algorithms have been proposed, which can be roughly divided into four groups: partitional, hierarchical, density-based, and subspace algorithms. Algorithms from the fourth group search for clusters in some lower-dimensional projection of the original data, and have been generally preferred when dealing with data that is high dimensional (Aggarwal *et al*., 2015). The motivation for this preference lies in the observation that having more dimensions usually leads to the so-called curse of dimensionality, where the performance of many standard machine-learning algorithms becomes impaired (Yu, 2013). This is mostly due to two pervasive effects: the empty space phenomenon and concentration of distances. The former refers to the fact that all high-dimensional data sets tend to be sparse, because the number of points required to represent any distribution grows exponentially with the number of dimensions. This leads to bad density estimates for high-dimensional data, causing difficulties for density-based approaches (Gnanabaskaran *et al*., 2011).

Corresponding author: **Pavithra, M.,**
Assistant Professor, Department C.S.E, Jansons Institute of Technology, Coimbatore, India.

The latter is a somewhat counterintuitive property of high-dimensional data representations. There are two main contributions of this paper. First, in experiments on synthetic data we show that hubness is a good measure of point centrality within a high-dimensional data cluster and that major hubs can be used effectively as cluster prototypes (Lance, 2014). In addition, we propose kernel mapping and collective neighbor clustering algorithms and evaluate their performance in various high-dimensional and semi-supervised data clustering tasks (Naveen, 2011). Clustering problem concerns the discovery of homogeneous groups of data according to a certain similarity measure. The task of clustering has been studied in statistics (Aggarwal, 2014), machine learning (Gnanabaskaran, 2011), bioinformatics (Yu, 2013), and more recently in databases (Singh *et al*., 2012). Clustering algorithms finds a partition of the points such that points within a cluster are more similar to each other than to points in different clusters (Karthikeyan, 2014). In traditional clustering each dimension is equally weighted when computing the distance between points. Most of these algorithms perform well in clustering low-dimensional datasets (Gnanabaskaran, 2011). However, in higher dimensional feature spaces, their performance and efficiency deteriorate to a greater extent due to the high dimensionality (Lance, 2014). Another difficulty we have to face when dealing with clustering is the dimensionality of data. In the clustering task, the overwhelming problem of high dimensionality presents a dual aspect. First, the presence of irrelevant attributes eliminates any hope on clustering.

Clustering suffers from the curse of dimensionality problem in high-dimensional spaces. In high dimensional spaces, it is highly likely that, for any given pair of points within the same cluster, there exist at least a few dimensions on which the points are far apart from each other (David *et al*., 2010). As a consequence, distance functions that equally use all input features may not be effective. Furthermore, several clusters may exist in different subspaces, comprised of different combinations of features. In many real world problems, in fact, some points are correlated with respect to a given set of dimensions, and others are correlated with respect to different dimensions (Karthikeyan, 2014). Each dimension could be relevant to at least one of the clusters. The problem of high dimensionality could be addressed by requiring the user to specify a subspace (i.e., subset of dimensions) for cluster analysis. However, the identification of subspaces by the user is an error-prone process. More importantly, correlations that identify clusters in the data are likely not to be known by the user (Guangtao *et al*., 2012). Indeed, we desire such correlations, and induced subspaces, to be part of the findings of the clustering process itself. An alternative solution to high dimensional settings consists in reducing the dimensionality of the input space. Traditional feature selection algorithms select certain dimensions in advance. Methods such as Principal Component Analysis (PCA) (or Karhunen– Loeve transformation) transform the original input space into a lower dimensional space by constructing dimensions that are linear combinations of the given features, and are ordered by non increasing variance (Hua-Liang, 2010). While PCA may succeed in reducing the dimensionality, with as major drawbacks. The new dimensions can be difficult to interpret, making it hard to understand clusters in relation to the original space. Furthermore, all global dimensionality reduction techniques (like PCA) are not effective in identifying clusters that may exist in different subspaces (Aggarwal, 2014).

RELATED WORK

Distributed Data Mining (DDM) is a dynamically growing area within the broader field of KDD. Generally, many algorithms for distributed data mining are based on algorithms which were originally developed for parallel data mining. In (Aggarwal, 2014), some state-of-the-art research results related to DDM are summarized. Whereas there already exist algorithms for distributed classification and association rules, there is a lack of algorithms for distributed clustering. In (Lance *et al*., 2014) the "collective hierarchical clustering algorithm" for vertically distributed data sets was proposed which applies single link clustering. In contrast to this approach, we concentrate in this paper on horizontally distributed data sets. In (Gnanabaskaran, 2011) the authors presented a technique for centroid-based hierarchical clustering for high-dimensional, horizontally distributed data sets by merging clustering hierarchies generated locally. Unfortunately, this approach can only be applied for distance-based hierarchical distributed clustering approaches, whereas our aim is to introduce a generally applicable approach. In (Naveen *et al*., 2011), density-based distributed clustering algorithms were presented which are based on the density-based partitioning clustering algorithm DBSCAN. The idea of these approaches is to determine suitable local objects representing several other local objects. Based on these representatives a global DBSCAN algorithm is carried out. These approaches are tailor-made for the density-based distributed clustering algorithm DBSCAN. The goal of this paper is to introduce an approach which is generally applicable to DDM. To get specific, we demonstrate the benefits of our approach for distributed clustering algorithms. In contrast to the above specific distributed clustering approaches, our approach is not susceptible to an increasing number of local clients (Sembiring *et al*., 2010). It does only depend on the overall allowed transmission cost, i.e. on the number of bytes we are allowed to transmit from the local clients to a server. In order to keep these transmission cost low, we introduce in the following section a suitable client-side approximation technique for describing high-dimensional feature vectors (Singh *et al*., 2012).

Dimensionality reduction is a technique that helps solving the high dimensionality problem and has been extensively studied and widely applied in text analysis (Aggarwal *et al*., 2014), face recognition (Yu *et al*., 2013), and microarray gene expression analysis (Singh *et al*., 2012) where data are usually expressed as vectors of high dimension. Dimensionality reduction is also the key technique for data compression that enables efficient information storage and retrieval (Lance *et al*., 2014), as well as for data visualization, where high-dimensional data are mapped to 2D or 3D spaces helping the user gain a qualitative understanding of the information (Gnanabaskaran, 2011). A dimensionality reduction technique finds low-dimensional structures of data hidden in high-dimensional observations. Feature selection (Naveen *et al*., 2011) and feature reduction (Sembiring *et al*., 2010) are two dimensionality reduction solutions. Feature selection reduces dimensionality by selecting a subset of existing features. Thus, the physical interpretation of each feature is preserved in the reduced space. However, in removing many features prior to learning from the data, information about the underlying data may be lost. Feature reduction reduces dimensionality by combining features with linear or nonlinear transformations (Mohamed *et al*., 2009). In (Aggarwal *et al*., 2015) authors are discuss very general techniques for projected clustering which

are able to construct clusters in arbitrarily aligned subspaces of lower dimensionality. The subspaces are specific to the clusters themselves. This definition is substantially more general and realistic than currently available techniques which limit the method to only projections from the original set of attributes (Gnanabaskaran *et al*., 2011). The generalized projected clustering technique may also be viewed as a way of trying to redefine clustering for high dimensional applications by searching for hidden subspaces with clusters which are created by inter-attribute correlations. In (Yu *et al*., 2013) authors used an application domains such as life sciences, e.g. molecular biology produce a tremendous amount of data which can no longer be managed without the help of efficient and effective data mining methods. One of the primary data mining tasks is clustering. However, traditional clustering algorithms often fail to detect meaningful clusters because of the high dimensional, inherently sparse feature space of most real-world data sets (Aggarwal *et al*., 2014). Nevertheless, the data sets often contain clusters hidden in various subspaces of the original feature space (Mohamed *et al*., 2009). A pre-processing step for traditional clustering algorithms, which detects all interesting subspaces of high-dimensional data containing clusters. For this purpose, we define a quality criterion for the interestingness of a subspace and propose an efficient algorithm called RIS (Ranking Interesting Subspaces) to examine all such subspaces. In (Gnanabaskaran, 2011) discussed the primary data mining tasks is clustering. However, traditional clustering algorithms often fail to detect meaningful clusters because most real-world data sets are characterized by a high dimensional, inherently sparse data space (Singh *et al*., 2012). Nevertheless, the data sets often contain interesting clusters which are hidden in various subspaces of the original feature space.

In (Lance *et al*., 2014) authors improved the conclusive evaluation and comparison is challenged by three major issues. First, there is no ground truth that describes the "true" clusters in real world data. Second, a large variety of evaluation measures have been used that reflect different aspects of the clustering result (Hua-Liang, 2010). Finally, in typical publications authors have limited their analysis to their favoured paradigm only, while paying other paradigms little or no attention. In (Naveen *et al*., 2011) authors proposed the dimensionality curse from the point of view of the distance metrics which are used to measure the similarity between objects.

The specifically examine the behaviour of the commonly used Lk norm and show that the problem of meaningfulness in high dimensionality is sensitive to the value of k. For example, this means that the Manhattan distance metric L1-norm is consistently more preferable than the Euclidean distance metric L2-norm for high dimensional data mining applications (10). Using the intuition derived from our analysis, we introduce and examine a natural extension of the Lk-norm to fractional distance metrics. In (Sembiring, 2010) authors considered a nearest neighbour search and many other numerical data analysis tools most often rely on the use of the Euclidean distance. When data are high dimensional, however, the Euclidean distances seem to concentrate; all distances between pairs of data elements seem to be very similar. Therefore, the relevance of the Euclidean distance has been questioned in the past, and fractional norms (Murkowski-like norms with an exponent less than one) were introduced to fight the concentration phenomenon (Guangtao Wang, 2012).

HIGH DIMENSIONAL DATA CLUSTERING

Clustering in high-dimensional spaces is a difficult problem which is recurrent in many domains, for example in image analysis. The difficulty is due to the fact that high-dimensional data usually live in different low-dimensional subspaces hidden in the original space (Aggarwal *et al*., 2012). This paper presents a family of Gaussian mixture models designed for high-dimensional data which combine the ideas of dimension reduction and parsimonious modelling. These models give rise to a clustering method based on the Expectation-Maximization algorithm which is called High-Dimensional Data Clustering (HDDC) (Gnanabaskaran, 2011). In order to correctly fit the data, HDDC estimates the specific subspace and the intrinsic dimension of each group. Our experiments on artificial and real datasets show that HDDC outperforms existing methods for clustering high-dimensional data. High-dimensional data, i.e., data described by a large number of attributes, pose specific challenges to clustering (Yu, 2013).

The so-called ‘curse of dimensionality’, coined originally to describe the general increase in complexity of various computational problems as dimensionality increases, is known to render traditional clustering algorithms ineffective (Lance *et al*., 2014). The curse of dimensionality, among other effects, means that with increasing number of dimensions, a loss of meaningful differentiation between similar and dissimilar objects is observed. As high-dimensional objects appear almost alike, new approaches for clustering are required (Naveen *et al*., 2011). Consequently, recent research has focused on developing techniques and clustering algorithms specifically for high-dimensional data. Still, open research issues remain. Clustering is a data mining task devoted to the automatic grouping of data based on mutual similarity. Each cluster group’s objects that are similar to one another, whereas dissimilar objects are assigned to different clusters, possibly separating out noise (Sembiring *et al*., 2010). In this manner, clusters describe the data structure in an unsupervised manner, i.e., without the need for class labels. A number of clustering paradigms exist that provide different cluster models and different algorithmic approaches for cluster detection (Mohamed, 2009). Common to all approaches is the fact that they require some underlying assessment of similarity between data objects. In this article, we provide an overview of the effects of high-dimensional spaces, and their implications for different clustering paradigms (Singh *et al*., 2012).

CURSE OF DIMENSIONALITY

The curse of dimensionality refers to various phenomena that arise when analyzing and organizing data in high-dimensional spaces (often with hundreds or thousands of dimensions) that do not occur in low-dimensional settings such as the three-dimensional physical space of everyday experience (Aggarwal *et al*., 2015). There are multiple phenomena referred to by this name in domains such as numerical analysis, sampling, combinatory, machine learning, data mining, and databases. The common theme of these problems is that when the dimensionality increases, the volume of the space increases so fast that the available data become sparse (Yu *et al*., 2013). This sparsity is problematic for any method that requires statistical significance. In order to obtain a statistically sound and reliable result, the amount of data needed to support the result often grows exponentially with the dimensionality (Gnanabaskaran, 2011).

Also, organizing and searching data often relies on detecting areas where objects form groups with similar properties; in high dimensional data, however, all objects appear to be sparse and dissimilar in many ways, which prevents common data organization strategies from being efficient (Lance *et al*., 2014). The use of the term "curse of dimensionality" in machine learning is related to the fact that one can easily imagine a target function (to be learned) that is very non-smooth, for example having an exponential number of modes (ups and downs), with respect to dimensionality (the number of scalar input variables) (Naveen *et al*., 2011). Imagine that in order to produce a good prediction, our learner needs to distinguish (produce a substantially different answer) between 10 different values of each of n variables (Sembiring *et al*., 2010). Then it may need to distinguish between 10^n different configurations of the input n-dimensional vector. With n easily in the hundreds, thousands or more, this is much more than the number of examples one can hope to gather (or even the number of atoms in the universe) (Singh *et al*., 2012). With most learning algorithms, and in particular with classical non-parametric learning algorithms (e.g. nearest-neighbour, Parzen, Gaussian kernel SVM, Gaussian kernel Gaussian Process, etc.) the learner will need to see at least one example for each of these many configurations (at least as many as necessary to cover all the variations of configurations of interest), in order to produce a correct answer around each of these configurations, one that is different from the target value required for other nearby configurations (Mohamed, 2009).

PROPOSED WORK

CLIQUE: THE CLASSICAL HIGH-DIMENSIONAL ALGORITHM

CLIQUE (Clustering in Quest), to find automatically subspace clustering of high dimensional numerical data. It locates clusters embedded in subspaces of high dimensional data without much user intervention to discern significant sub clusters (Aggarwal *et al*., 2005). CLIQUE first partitions its numerical space into units for its grid structure. CLIQUE divides the d-dimensional data space into md non-overlapping rectangular units. A d-dimensional data point, v, is considered in a unit, u, if the value of v in each attribute, is greater than or equal to the left boundary of that attribute in u and less than the right boundary of that attribute in u (Gnanabaskaran, 2011). The selectivity of a unit is defined to be the fraction of total data points in the unit. Only units whose selectivity is greater than a parameter τ are viewed as dense and retained. The definition of dense units applies to all subspaces of the original d-dimensional space (Yu *et al*., 2013). CLIQUE prunes the pool of candidates, only keeping the set of dense units to form the candidate units in the next level of the dense unit generation algorithm (Naveen *et al*., 2011). To prune the candidates, all the subspaces are sorted by their coverage, i.e., the fraction of the database that is covered by the dense units in it. CLIQUE then forms clusters from the remaining candidate units (Lance *et al*., 2014). Two p-dimensional units u1, u2 are connected if they have a common face or if there exists another p-dimensional unit us such that u1 is connected to us and u2 is connected to us. A cluster is a maximal set of connected dense units in p-dimensions (Gnanabaskaran, 2011).

$$\mu_I(i) = \left\lceil \frac{\sum_{1\le j\le i} x_{S_j}}{i} \right\rceil \; ; \; \mu_P(i) = \left\lceil \frac{\sum_{i+1\le j\le n} x_{S_j}}{n-i} \right\rceil$$

$$CL(i) = \log_2(\mu_I(i)) + \sum_{1\le j\le i} \log_2(|x_{S_j} - \mu_I(i)|) + \log_2(\mu_P(i)) + \sum_{i+1\le j\le n} \log_2(|x_{S_j} - \mu_P(i)|)$$

VARIANTS OF CLIQUE

There are two aspects of the CLIQUE algorithm that can be improved. The first one is the criterion for the subspace selection. The second is the size and resolution of the grid structure (Sembiring *et al*., 2010). The former is addressed by the ENCLUS algorithm by using entropy as subspace selection criterion. The latter is addressed by the MAFIA algorithm by using adaptive grids for fast subspace clustering (Mohamed, 2009).

insert into C_k
select $u_1.[l_1, h_1), u_1.[l_2, h_2), \ldots, u_1.[l_{k-1}, h_{k-1}), u_2.[l_{k-1}, h_{k-1})$
from $D_{k-1}\ u_1,\ D_{k-1}\ u_2$
where $u_1.a_1 = u_2.a_1,\ u_1.l_1 = u_2.l_1,\ u_1.h_1 = u_2.h_1,$
$u_1.a_2 = u_2.a_2,\ u_1.l_2 = u_2.l_2,\ u_1.h_2 = u_2.h_2, \ldots,$
$u_1.a_{k-2} = u_2.a_{k-2},\ u_1.l_{k-2} = u_2.l_{k-2},\ u_1.h_{k-2} = u_2.h_{k-2},$
$u_1.a_{k-1} < u_2.a_{k-1}$

ENCLUS: ENTROPY-BASED APPROACH

The algorithm ENCLUS (Entropy-based Clustering) (Naveen *et al*., 2011) is an adaptation of the CLIQUE that uses a different, entropy-based criterion for subspace selection. Rather than using the fraction of total points in a subspace as a criterion to select subspaces, ENCLUS uses an entropy criterion and only those subspaces spanned by attributes (Singh *et al*., 2012). An analogous monotonicity condition or Apriori property also exists in terms of entropy. If a p-dimensional subspace has low entropy, then so does any (p−1)-dimensional projections of this subspace. A significant limitation of ENCLUS is its extremely high computational cost, especially in terms of computing the entropy of subspaces (Mohamed, 2009). However, this cost also yields the benefit that this approach has increased sensitivity to detect clusters especially extremely dense small ones (Aggarwal *et al*., 2005).

$$\begin{aligned} & H(X_1, \ldots, X_{k-1}) \\ \le\ & H(X_1, \ldots, X_{k-1}) + H(X_k | X_1, \ldots, X_{k-1}) \text{ (non-negativity)} \\ =\ & H(X_1, \ldots, X_k) \\ <\ & \omega \end{aligned}$$

Algorithm 2 ENCLUS_INT(ω, ϵ')
1 $k = 1$
2 Let C_k be all one dimensional subspaces.
3 For each subspace $c \in C_k$ do
4 $f_c(\cdot)$ = cal_density(c)
5 $H(c)$ = cal_entropy($f_c(\cdot)$)
6 If $H(c) < \omega$ then
7 If $interest\_gain(c) > \epsilon'$ then
8 $I_k = I_k \cup c$.
9 else
10 $NI_k = NI_k \cup c$.
11 End For
12 C_{k+1} = candidate_gen($I_k \cup NI_k$)
13 If $C_{k+1} = \emptyset$, go to step 16.
14 $k = k + 1$
15 Go to step 3.
16 $Result = \bigcup_{\forall k} I_k$

MAFIA: ADAPTIVE GRIDS IN HIGH DIMENSIONS

MAFIA (Merging of Adaptive Finite Intervals) proposed by Goil *et al*. (Aggarwal *et al*., 2005) is a descendant of CLIQUE. Instead of using a fixed size cell grid structure with an equal number of bins in each dimension, MAFIA constructs adaptive grids to improve subspace clustering and also uses parallelism on a shared-nothing architecture to handle massive data sets (Yu *et al*., 2013). MAFIA proposes an adaptive grid of bins in each dimension. Then using an Apriori algorithm, dense intervals are merged to create clusters in the higher dimensional space (Gnanabaskaran, 2011). The adaptive grid is created by partitioning each dimension independently based on the distribution (i.e., the histogram) observed in that dimension, merging intervals that have the same observed distribution, and pruning those intervals with low density (Naveen *et al*., 2011). This pruning during the construction of the adaptive grid reduces the overall computation of the clustering step (Lance *et al*., 2014).

ALGORITHM FOR MAFIA ALGORITHM

- Do one scan of the data to construct adaptive grids in each dimension.
- Compute the histograms by reading blocks of data into memory using bins.
- Using the histograms to merge bins into a smaller number of adaptive variable-size bins, where adjacent bins with similar histogram values are combined to form larger bins. The bins that have low density of data are pruned.
- Select bins that are α-times (α is a parameter called the cluster dominance factor) more densely populated than average as p (p = 1 now) candidate dense units (CDUs).
- Iteratively scan data for higher dimensions, and construct new p-CDU from two (p−1) CDUs if they share any (p−2)-face, and merge adjacent CDUs into clusters.
- Generate minimal DNF expressions for each cluster

OPTIGRID: DENSITY-BASED OPTIMAL GRID PARTITIONING

Hinneburg and Keim proposed OptiGrid (Optimal GRID-Clustering) (Guangtao *et al*., 2012) to address several aspects of the "curse of dimensionality": noise, scalability of the grid construction, and selecting relevant attributes by optimizing the density function over the data space. OptiGrid uses density estimations to determine the centers of clusters as the clustering was done for the DENCLUE algorithm (Aggarwal *et al*., 2014).

$$H(x) = \begin{cases} 1 & if \quad \sum_{i=1}^{d} w_i x_j \geq 1, \\ 0 & otherwise. \end{cases}$$

A cluster is a region of concentrated density centered around a strong density attractor or local maximum of the density function with density above the noise threshold (Yu *et al*., 2013). Clusters may also have multiple centers if the centers are strong density attractors and there exists a path between them above the noise threshold (Naveen *et al*., 2011). By recursively partitioning the feature space into multidimensional grids, OptiGrid creates an optimal grid-partition by constructing the best cutting hyper planes of the space (Lance *et al*., 2014).

$$x \in S, c(x) = \sum_{i=1}^{k} 2^i \cdot H_i(x).$$

These cutting planes cut the space in areas of low density (i.e. local minima of the density function) and preserve areas of high density or clusters, specifically the cluster centers (i.e. local maxima of the density function) (Sembiring *et al*., 2010). The cutting hyper planes are found using a set of contracting linear projections of the feature space. The contracting projections create upper bounds for the density of the planes orthogonal to them (Singh *et al*., 2012). Namely, for any point, x, in a contracting projection, P, then for any point y such that P(y) = x, the density of y is at most the density of x (10).

$$\hat{f}^D = \frac{1}{nh} \sum_{i=1}^{n} KD(\frac{x - x_i}{h}),$$

ALGORITHM FOR OPTIGRID ALGORITHM

INPUT: data set D, q, min cut score

- Determine a set of contracting projections P = {P0, P1... Pk} and calculate all the projections of the data set D: Pi (D), i = 1, 2... K;
- Initialize a list of cutting planes BEST CUT □ Φ, CUT ⇐ Φ;
- for i=0 to k do
- CUT ⇐best local cuts Pi (D);
- CUT SCORE ⇐Score best local cuts Pi (D);
- Insert all the cutting planes with a score ≥ min cutscore into BEST CUT;
- if BEST CUT = Φ then
- return D as a cluster;
- else
- Select the q cutting planes of the highest score from BEST CUT and construct a multidimensional grid G using the q cutting planes;
- Insert all data points in D into G and determine the highly populated grid cells in G; add these cells to the set of clusters C;
- Refine C;
- for all clusters Ci in C do
- Do the same process with data set Ci;
- end for
- end if
- end for

O-CLUSTER: A SCALABLE APPROACH

Milenova *et al*. proposed a O-cluster (Orthogonal partitioning Clustering) to address three limitations of OptiGrid: scalability in terms of data relative to memory size, lack of clear criterion to determine if a cutting plane is optimal or not, and sensitivity to threshold parameters for noise and cut plane density (Yu *et al*., 2013). O-Clusters address the first limitation by using a random sampling technique on the original data and a small buffer size. Only partitions that are not resolved (i.e., ambiguous) have data points maintained in the buffer (Lance *et al*., 2014). As a variant of OptiGrid, O-Cluster uses an axis-parallel partitioning strategy to locate high density areas in the data (Singh *et al*., 2012). To do so, O-Cluster uses contracting projections, but also proposes the use of a statistical test to validate the quality of a cutting plane (Naveen *et al*., 2011).

The statistical test checks for statistical significance between the difference in the density of the peaks and a valley when the valley separates the two peaks using a standard $\chi 2$ test (Sembiring *et al*., 2010). If statistical significance is found, the cutting plane would then be through such a valley. O-Cluster is also a recursive method. After testing the splitting points for all projections in a partition, the optimal one is chosen to partition the data. The algorithm then searches for cutting planes in the new partitions (Mohamed, 2009).

ALGORITHM FOR O-CLUSTER: A SCALABLE APPROACH

- Load data buffer.
- Compute histograms for active partitions.
- Find "best" splits for active partitions.
- Flag ambiguous and "frozen" partitions.
- Split active partitions.
- Reload buffer.

EXPERIMENTS

In this section, we empirically demonstrate that our proposed high dimensional data clustering algorithm is both efficient and effective.

DATASETS

The data sets used in our experiments include six UCI data sets1. Here is some basic information of those data sets. Table 5 summarizes the basic information of those data sets.

- **Balance:** This data set was generated to model psychological experimental results. There are totally 625 examples that can be classified as having the balance scale tip to the right, tip to the left, or be balanced.
- **Iris:** This data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant.
- **Ionosphere:** It is a collection of the radar signals belonging to two classes. The data set contains 351 objects in total, which are all 34-dimensional.
- **Soybean:** It is collected from the Michalski's famous soybean disease databases, which contains 562 instances from 19 classes.

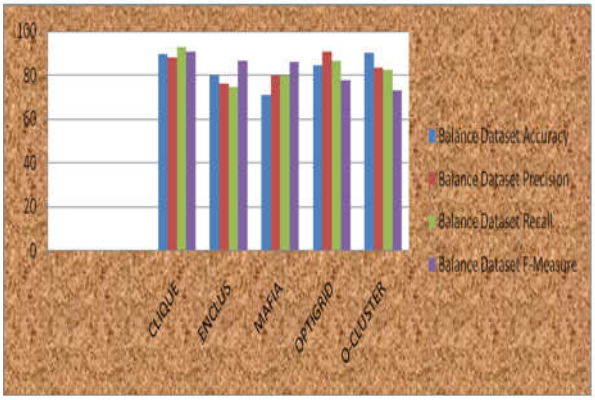
EXPERIMENTAL RESULTS

Datasets	Size	Classes	Dimensions
Balance	625	3	4
Iris	150	3	4
Ionosphere	351	2	34
Soybean	562	19	35

BALANCE DATASET RESULTS

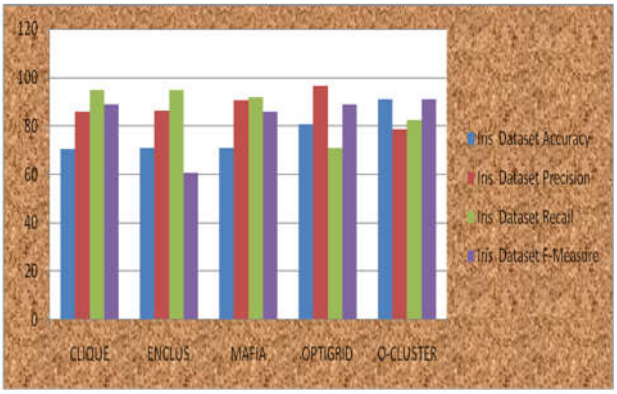
The above graph shows that performance of Balance dataset. The Accuracy of O-CLUSTER algorithm is 90.07 which is higher when compare to other four (CLIQUE, ENCLUS, MAFIA, OPTIGRID) algorithms. The Precision of OPTIGRID algorithm is 90.67 which is higher when compare to other four (CLIQUE, ENCLUS, MAFIA, O-CLUSTER) algorithms. The Recall of CLIQUE algorithm is 92.77 which is higher when compare to other four (ENCLUS, MAFIA, O-CLUSTER, OPTIGRID) algorithms. The F-Measure of CLIQUE algorithm is 90.89 which is higher when compare to other four (ENCLUS, MAFIA, O-CLUSTER, OPTIGRID) algorithms.

Balance Dataset				
Algorithm	Accuracy	Precision	Recall	F-Measure
CLIQUE	89.45	87.91	92.77	90.89
ENCLUS	79.91	76.08	74.78	86.56
MAFIA	70.92	79.67	79.89	85.78
OPTIGRID	84.67	90.67	86.78	77.67
O-CLUSTER	90.07	83.66	82.33	72.88



IRIS DATASET RESULTS

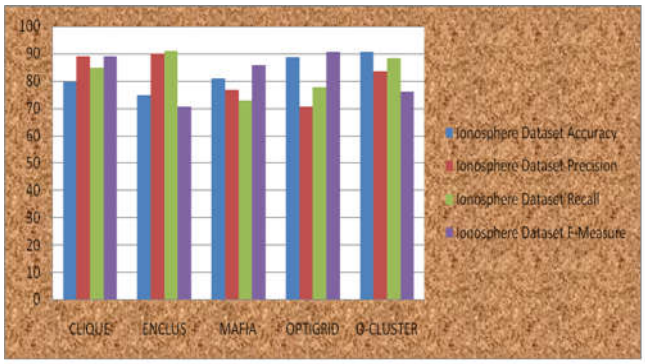
Iris Dataset				
Algorithm	Accuracy	Precision	Recall	F-Measure
CLIQUE	70.45	85.91	94.77	88.89
ENCLUS	70.91	86.08	94.78	60.56
MAFIA	70.92	90.67	91.89	85.78
OPTIGRID	80.67	96.67	70.78	88.67
O-CLUSTER	90.78	78.76	82.54	90.89



The above graph shows that performance of Iris dataset. The Accuracy of O-CLUSTER algorithm is 90.78 which is higher when compare to other four (CLIQUE, ENCLUS, MAFIA, OPTIGRID) algorithms. The Precision of OPTIGRID algorithm is 96.67 which is higher when compare to other four (CLIQUE, ENCLUS, MAFIA, O-CLUSTER) algorithms. The Recall of ENCLUS algorithm is 94.78 which is higher when compare to other four (CLIQUE, O-CLUSTER, MAFIA, OPTIGRID) algorithms. The F-Measure of O-CLUSTER algorithm is 90.89 which is higher when compare to other four (CLIQUE, ENCLUS, MAFIA, OPTIGRID) algorithms.

IONOSPHERE DATASET RESULTS

Ionosphere Dataset				
Algorithm	Accuracy	Precision	Recall	F-Measure
CLIQUE	79.45	88.91	84.77	88.89
ENCLUS	74.91	90.08	90.78	70.56
MAFIA	80.98	76.67	72.89	85.78
OPTIGRID	88.67	70.67	77.78	90.67
O-CLUSTER	90.56	83.45	88.34	75.89

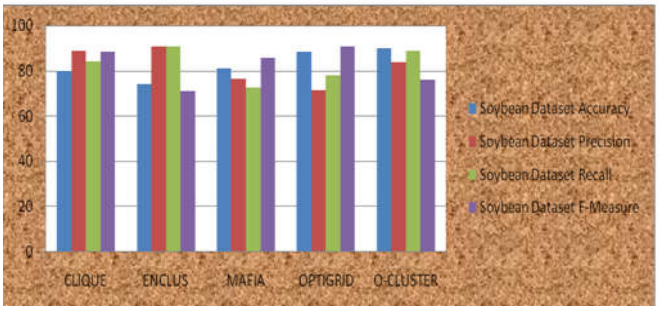


The above graph shows that performance of Ionosphere dataset. The Accuracy of O-CLUSTER algorithm is 90.56 which is higher when compare to other four (CLIQUE, ENCLUS, MAFIA, OPTIGRID) algorithms. The Precision of ENCLUS algorithm is 90.08 which is higher when compare to other four (CLIQUE, O-CLUSTER, MAFIA, OPTIGRID) algorithms. The Recall of ENCLUS algorithm is 90.78 which is higher when compare to other four (CLIQUE, O-CLUSTER, MAFIA, OPTIGRID) algorithms. The F-Measure of OPTIGRID algorithm is 90.67 which is higher when compare to other four (CLIQUE, ENCLUS, MAFIA, O-CLUSTER) algorithms.

SOYBEAN DATASET RESULTS

The above graph shows that performance of Soybean dataset. The Accuracy of O-CLUSTER algorithm is 90.08 which is higher when compare to other four (CLIQUE, ENCLUS, MAFIA, OPTIGRID) algorithms. The Precision of ENCLUS algorithm is 90.89 which is higher when compare to other four (CLIQUE, O-CLUSTER, MAFIA, OPTIGRID) algorithms.

Soybean Dataset				
Algorithm	Accuracy	Precision	Recall	F-Measure
CLIQUE	79.89	88.65	84.23	88.34
ENCLUS	74.03	90.89	90.67	71.23
MAFIA	81.08	76.32	72.45	85.9
OPTIGRID	88.54	71.32	77.89	90.56
O-CLUSTER	90.08	83.78	88.78	75.9



The Recall of ENCLUS algorithm is 90.67 which is higher when compare to other four (CLIQUE, O-CLUSTER, MAFIA, OPTIGRID) algorithms. The F-Measure of OPTIGRID algorithm is 90.56 which is higher when compare to other four (CLIQUE, ENCLUS, MAFIA, O-CLUSTER) algorithms.

Conclusion

The purpose of this article is to present a comprehensive classification of different clustering techniques for high dimensional data. Clustering high dimensional data sets is a ubiquitous task. The incosent growth in the fields of communication and technology, there is tremendous growth in high dimensional data spaces. It study focuses on issues and major drawbacks of existing algorithms (Aggarwal *et al*., 2005). As the number of dimensions increase, many clustering techniques begin to suffer from the curse of dimensionality, de-grading the quality of the results. In high dimensions, data becomes very sparse and distance measures become increasingly meaningless (Gnanabaskaran, 2011). This problem has been studied extensively and there are various solutions, each appropriate for different types of high dimensional data and data mining procedures (Yu *et al*., 2013). As with any clustering techniques, finding meaningful and useful results depends on the selection of the appropriate clustering technique (Lance *et al*., 2014).

In order to do this, one must understand the dataset in a domain specific context in order to be able to best evaluate the results from various approaches. From the above discussion it is observed that the current techniques will suffer with many problems (Naveen *et al*., 2011). To improve the performance of the data clustering in high dimensional data, it is necessary to perform research in the areas like dimensionality reduction, redundancy reduction in clusters and data labelling. The feature selection is a complex problem studied by many researchers all over the world (Singh *et al*., 2012). Complexity is due to finding a voluminous amounts of High Dimensional data, contains irrelevant or redundant features which causes difficulties in storage and retrieval. The feature subset selection algorithm for high dimensional data works based on the clusters that contains features where each cluster treated as single feature and hence dimensionality of data is drastically reduces. We then used this as a goodness-of-fit measure in the context of subspace clustering (Sembiring *et al*., 2010). The resulting subspace clustering method achieved state-of-the art clustering accuracy and speed on both simulated and real datasets (Aggarwal *et al*., 2014). Whilst this works well for certain applications such as images of faces and motion tracking, it may be desirable to develop a more general framework for identifying clusters which do not necessarily lie in linear subspaces (Yu *et al*., 2013). Recently, the field of multiple manifold clustering has emerged where the aim is to find clusters which lie in non-linear manifolds of which subspace clustering is a special case (Lance *et al*., 2014). We proposed a new approach to multi-view clustering which takes a step towards consolidating supervised and unsupervised learning in the multi-view setting. This allows us to model more complicated dependencies between the views than the usual conditional independence assumption allows (Gnanabaskaran, 2011). Our approach can be viewed as an extension of subspace clustering in two views and so carries with it the same benefits of subspace clustering compared to geometric-distance based clustering (Sembiring *et al*., 2010). The field of multi-view clustering where there is a predictive relationship between the views has not been well developed and so our work represents a significant and novel contribution which consolidates supervised and unsupervised approaches to multi-view learning (Mohamed, 2009).

FUTURE WORK

Our proposed baseline includes multiple aspects for a fair comparison not only in evaluation studies: First, a common open source framework with baseline implementations for a fair comparison of different algorithms (Aggarwal *et al*., 2014). Second, a broad set of evaluation measures for clustering quality comparison. Third, a baseline of evaluation results for both real world and synthetic data sets with given parameter settings for repeatability (Yu *et al*., 2013). All of

this can be downloaded from our website for further research, comparison or repeatability. Recent approaches of these paradigms enhanced the quality and efficiency, however, could reach top results only in few cases. One downside of the system is the heavy processing loads especially for high dimensions that prompted us to down sample some data (Aggarwal *et al*., 2005). One of the goals in future works will be to improve this performance through parallel implementations or the use of GPUs (Gnanabaskaran, 2011).

In addition, we plan to further explore the various repeating patterns found during the animated transition, such as the compression/expansion connection, to better understand their connection with the underlying high-dimensional structure (Naveen *et al*., 2011). It deals with removing of irrelevant and redundant data or feature set that leads to provide high accurate feature as per required target class (Lance *et al*., 2014). In future work, we plan to address the problem of evaluating the quality of clustering's in different subspaces. One approach is to choose clusters that maximize the ratio of cluster density over expected density for clustering's with the same dimensionality (Singh *et al*., 2012). The results reported for these applications show that use of the method is promising in various applications, including dominant and deviant pattern detection, collaborative filtering, clustering, bounded error compression, and classification (Sembiring *et al*., 2010). The method can also be extended beyond binary attributed datasets to general discrete positive valued attribute sets. The techniques discussed in this paper extend the applicability of outlier detection techniques to high dimensional problems; such cases are most valuable from the perspective of data mining applications (David *et al*., 2010). For the future work, we plan to explore different types of correlation measures, and study some formal properties of feature space. In feature we are going to classify the high dimensional data (Karthikeyan *et al*., 2014). Future enhancement of our algorithm will allow variable number of clusters at each iteration. Our algorithm needs to be more scalable for huge data set (Hua-Liang, 2010). We had saved time by using index structure and minimizing range queries by focusing on subspaces which are irrelevant and a subspace which is very similar to it (Aggarwal *et al*., 2014). Time can be saved in executing a single range query. In future the performance of clustering is improved by considering the time and iteration factors (Yu *et al*., 2013).

REFERENCES

Aggarwal, C., Han, J., Wang, J. and Yu, P. 2014. "A Framework for Projected Clustering of High Dimensional Data Streams", In VLDB Conference, 2014.

Aggarwal, C., Han, J., Wang, J. and Yu, P. 2015. "On High Dimensional Projected Clustering of Data Streams*", Data Mining and Knowledge Discovery Journal*, 10(3), pp. 251–273.

David L. Donoho, 2010. "High Dimensional Data Analysis: The Curses and Blessings of Dimensionality," American Math. Society Conference: Mathematical Challenges of the 21st Century, Los Angeles, CA, August, 6-11.

Gnanabaskaran A. and Duraiswamy K. 2011. "An Efficient Approach to Cluster High Dimensional Spatial Data Using K, Mediods Algorithm," European *Journal of Scientific Research*, vol. 49 no. 4, pp. 617,624.

Guangtao Wang, Qinbao Song, Baowen Xu, Yuming Zhou, 2012. "Selecting feature subset for high dimensional data via the propositional FOIL rules"-Elsevier-.

Hua-Liang Wei and Stephen A. 2010. Billings,"Feature subset selection and Ranking for Data Dimensionality Reduction", IEEE Transactions on pattern analysis and machine intelligence, Vol. 29, N0.1.

Karthikeyan.P,Saravanan, P, Vanitha, E. 2014. "High Dimensional Data Clustering using FAST Cluster Based feature selection, Journal of engineering research and application Vol.4,pp.65-71.

Lance P., Ehtesham H., and Huan L. 2014. "Subspace Clustering for High Dimensional Data: A Review," ACM SIGKDD Explorations Newsletter, vol. 6, no. 1, pp. 90,105.

Mohamed B. and Shergrui W. 2009. "Mining Projected Clusters in High, Dimensional Spaces," IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 4, pp. 507,522.

Naveen K., Naveen G., and Veera R. 2011. "Partition Algorithms, A Study and Emergence of Mining Projected Clusters in High, Dimensional Dataset," the International Journal of Computer Science and Telecommunications, vol. 2, no. 4, pp. 34,37.

Sembiring R., Zain J., and Abdullah E. 2010. "Clustering High Dimensional Data using Subspace and Projected Clustering Algorithms," the *International Journal of Computer Science & Information Technology*, vol. 2, no. 4, pp. 162, 170.

Singh V., Sahoo L., and Kelkar A. 2012. "Mining Subspace Clusters in High Dimensional Data," the International Journal of Recent Trends in Engineering and Technology, vol. 3, no. 1, pp. 118,112.

Yu L. and Liu, H. 2013. "Efficiently Handling Feature Redundancy in High-Dimensional Data,"Proc. Ninth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '03),pp. 685-690.

****