http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 10 (6): 1092-1105, 2011
ISSN 1812-5638 / DOI: 10.3923/itj.2011.1092.1105

Efficient Clustering for High Dimensional Data: Subspace Based Clustering and Density Based Clustering

Singh Vijendra
Department of Computer Science and Engineering, Faculty of Engineering and Technology,
Mody Institute of Technology and Science, Lakshmangarh, Sikar, Rajasthan, India

Abstract: Finding clusters in a high dimensional data space is challenging because a high dimensional data space has hundreds of attributes and hundreds of data tuples and the average density of data points is very low. The distance functions used by many conventional algorithms fail in this scenario. Clustering relies on computing the distance between objects and thus, the complexity of the similarity models has a severe influence on the efficiency of the clustering algorithms. Especially for density-based clustering, range queries must be supported efficiently to reduce the runtime of clustering. The density-based clustering is also influenced by the density divergence problem that affects the accuracy of clustering. If clusters do not exist in the original high dimensional data space, it may be possible that clusters exist in some subspaces of the original data space. Subspace clustering algorithms localize the search for relevant dimensions allowing them to find clusters that exist in multiple, possibly overlapping subspaces. Subspace clustering algorithms identifies such subspace clusters. But for clustering based on relative region densities in the subspaces, density based subspace clustering algorithms are applied where the clusters are regarded as regions whose densities are relatively high as compared to the region densities in a subspace. This study presents a review of various subspaces based clustering algorithms and density based clustering algorithms with their efficiencies on different data sets.

Key words: Feature selection, Subspace clustering, density based clustering, high dimensional data

INTRODUCTION

Clustering is one of the major data mining tasks and aims at grouping the data objects into meaningful classes (clusters) such that the similarity of objects within clusters is maximized and the similarity of objects from different clusters is minimized (Kaufman and Rousseeuw, 1990). Cluster analysis is one of the main tools for exploring the underlying structure of a data set. Clustering finds important applications in a wide variety of disciplines including remote sensing, pattern recognition, image processing and computer vision (Han and Kamber, 2001). The prime objective of a clustering technique is to partition a given data set consisting of N-dimensional points or vectors into a fixed number of L clusters. Traditionally, clustering is considered to be a process that partitions the data points into mutually exclusive groups or clusters such that data points in the same cluster are more similar to each other than to data points in other clusters (Arora *et al*., 2009). The dissimilarity between a pair of data points is usually measured by a distance metric defined on the differences between the values of their attributes (dimensions). Traditional clustering algorithms (Macqueen, 1967; Cutting *et al*., 1992; Defays, 1977; Frank and Roberto, 1994; Hartigan, 1975; Amir and Lipika, 2007; Sibson, 1973; Ranjan and Khalil, 2007) use all the attributes in the data to compute the distances. The curse of dimensionality makes the clustering task very difficult when the data space contains a large number of attributes. The large number of attributes makes it computationally infeasible to use all the attributes to find the clusters. Besides, not all the attributes are useful for the clustering task. The irrelevant attributes cause the average density for a cluster in any neighborhood of the data space to be low which makes it impossible to find any meaningful clusters using the traditional clustering algorithms in full-dimensional space.

In order to overcome the limitations of traditional clustering algorithms, some attempts have been made to use genetic algorithms for clustering data sets. Tseng and Yang (2001) proposed a genetic algorithm based approach for the clustering problem. Bandyopadhyay and Maulik (2002) applied the variable string length genetic algorithm with the real encoding of the coordinates of the cluster centers in the chromosome to the clustering problem. Bandyopadhyay and Saha (2007) proposed an evolutionary clustering technique that uses a new point symmetry-based distance measure. Vijendra *et al*. (2010) proposed a Genetic Clustering Algorithm (GCA) that finds a globally optimal partition of a given data sets into a

specified number of clusters. A genetic algorithm with chromosome reorganize (GACR) is introduced by Singh *et al.* (2011) to enhance the performance of clustering. In GACR, the degeneracy of chromosome is effectively removed which makes the evolution process converge fast.

A common approach to cope with the curse of dimensionality problem for mining tasks is to reduce the data dimensionality by using the techniques of feature transformation and feature selection. The dimensionality reduction (Roweis and Saul, 2000; Gao *et al.*, 2009) focuses on the problem of reducing the number of features under assumption that not all feature dimensions, in the original space, are relevant to the given learning tasks. Dimensionality reduction provides an effective way to reduce the feature size and improves the learning efficiency and effectiveness. But different groups of points may be clustered in different subspaces; a significant amount of research has been elaborated upon subspace clustering (Chu *et al.*, 2009; Singh *et al.*, 2010; Chen *et al.*, 2011) which aims at discovering clusters embedded in any subspace of the original feature space. Subspace clustering therefore aims at detecting clusters in any possible attribute combination.

Density-based approaches are very popular to determine clusters in subspace. As the number of subspace projections is exponentially with the number of dimensions, subspace clustering methods have a tremendous need for efficient density-based methods. Density-based clustering algorithms, search for dense subspaces. A dense subspace is defined by a radius of maximum distance from a central point and it has to contain many objects according to a threshold criterion. Density-based clustering (Ester *et al.*, 1996; Hinneburg and Keim, 1998; Friedman and Meulman, 2004; Yousria *et al.*, 2009; Singh and Trikha, 2011) defines clusters as regions with a high density of objects separated by regions with low density. The main process is to explore these two region types. A common technique is to partition the data set into non overlapping cells. The cells with a high density of data points are supposed to be cluster centers whereas the boundaries between clusters fall into the regions with low density.

There are a number of excellent reviews of clustering techniques available. Jain *et al.* (1999) published a review on data clustering. Kolatch (2001) presented a survey of clustering algorithms for spatial databases and a survey of clustering data mining technique. Garg and Jain (2006) performed a study on clustering algorithms based on partition and variation of k-means algorithm. One of comprehensive review was published by Parsons *et al.* (2004) on subspace clustering for high dimensional data. Xu and Wunsch (2005) also presented a survey of clustering algorithms. Recently, Sun *et al.* (2008) published another survey on clustering algorithms research. Velmurugan and Santhanam (2011) explored the behavior of some of the partition based clustering algorithms and their basic approaches with experimental results in their survey of partition based clustering algorithms in data mining: an experimental approach.

PROBLEM WITH CLUSTERING HIGH DIMENSIONALITY

Clustering high dimensional data is usually a difficult task. In fact, most traditional clustering algorithms tend to break down when applied to high dimensional feature spaces. Another difficulty we have to face when dealing with clustering is the dimensionality of data. The objects could be described by hundreds of attributes and these results in high dimensional datasets. In clustering, the overwhelming problem of high dimensionality presents a dual aspect. First, the presence of irrelevant attributes eliminates any hope on clustering tendency, because such features cause the algorithm search for clusters where there are no ones. This also happens with low dimensional data, but the likelihood of presence of irrelevant features and their number grow with dimension. The second problem is the so called Curse of dimensionality. For clustering this means that cluster do not show across all attributes as they are hidden by irrelevant attributes or blurred by noise. Clustering methods are typically either based on distances (like partitioning and hierarchical clustering) or on densities (like density-based methods). Beyer *et al.* (1999) studied the effects of high dimensions on the nearest neighbor d_{min} (o) and the farthest neighbor d_{max}(o) of an object o in detail. They have proven the following equation for different distributions:

$$\forall \epsilon \geq 0: \lim_{dim \to \infty} P\,(dmax\,(o) < (1 + \epsilon)\,dmin\,(o)) = 1 \quad (1)$$

This statement formalizes that with growing dimensionalities (dim) the distance to the nearest neighbor is nearly equal to the distance to the farthest neighbor (distances become more and more similar). Consequently, clustering methods based on distance functions have problems to extract meaningful patterns in high dimensional spaces as they either cluster only one object (the nearest neighbor) or nearly the complete data set (the farthest neighbor). Figure 1 shows that clusters are embedded in different subspaces of high-dimensional data sets.

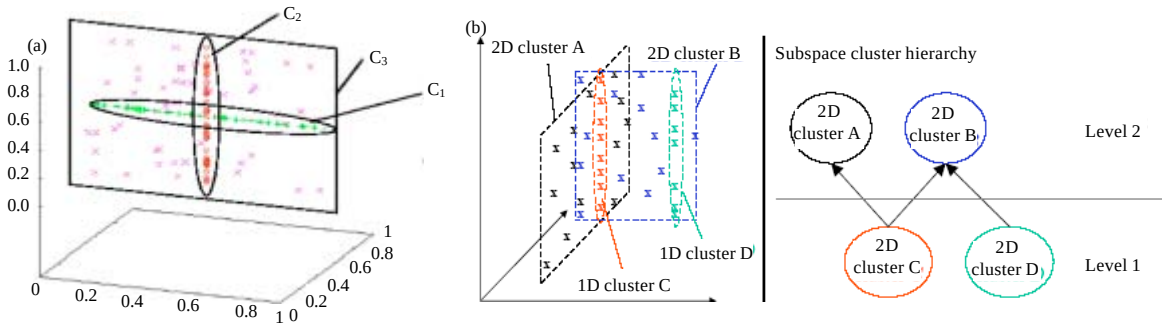

Fig. 1: (a) Subspace clusters and (b) Hierarchies of subspace clusters

Densities also suffer from the curse of dimensionality. Sibson (1973) described an effect of higher dimensions on density distributions: 99% of the mass of a ten-dimensional normal distribution is at points whose distance from the origin is greater than 1.6. This effect is directly opposite in lower dimensional spaces: 90% of the objects have a distance of less than 1:6 from the origin regarding a one-dimensional distribution. Density-based clustering methods hence have problems to determine the density of a region as the objects are scattered over the data space.

There are two traditional ways to tackle the problem of high dimensionality. The first one consists in a variety of techniques to perform a dimensionality reduction before the clustering process, so we can work on a dataset ideally equivalent to the original one but with a lower dimensionality (and sometimes with a lower level of noise). The second way is known as subspace clustering and density based clustering which is a special class of clustering algorithms that try to circumvent high dimensionality by building clusters in appropriate subspaces of the original feature space.

DIMENSIONALITY REDUCTION

Techniques for reduction of the dimensionality (Schott, 1993; Roweis and Saul, 2000; Gao *et al*., 2009) in datasets could be divided in three classes: feature extraction (also known as feature transformation), feature selection and, more recently, feature clustering. Feature extraction consists in applying a mapping from the multidimensional space to a space of fewer dimensions. This means that the original feature space is transformed by creating new features from combinations of the original ones. Feature selection methods, indeed, select only the most relevant of the dimensions from a dataset to reveal groups of objects that are similar on only a subset of their attributes. Feature clustering is a more recent class of methods for the reduction of dimensionality. It consists in performing the clustering of the feature set on a per-objects basis, i.e., it is a clustering of the transposed data matrix.

Feature extraction: Feature extraction is commonly used on high dimensional datasets. These methods include techniques such as PCA (Bishop, 2006), Singular Value Decomposition (SVD) or Sammon's non linear mapping (Sammon, 1969) which is often used in data mining and image analysis. However, PCA (Jolliffe, 2002) is a linear technique, i.e., it only takes into account linear dependences between variables. Recently, many non-linear techniques have been proposed such as Kernel PCA (Scholkopf *et al*., 1998) non-linear PCA (Girard, 2000) and neural networks based techniques (Roweis and Saul, 2000) recently investigate the situation when the data are not drawn from Gaussian distributions and use Laplacian distribution (or L1 distribution) a heavy-tailed distribution, to obtain a robust multivariate L1 PCA (L1-PCA) method (Gao *et al*., 2009) which is less sensitive to outliers. Feature extraction is often a preprocessing step, allowing the clustering algorithm to use just a few of the newly created features. A few clustering methods have incorporated the use of such transformations to identify important features and iteratively improve their clustering. While often very useful, these techniques do not actually remove any of the original attributes from consideration. Thus, information from irrelevant dimensions is preserved, making these techniques ineffective at revealing clusters when there are large numbers of irrelevant attributes that mask the clusters with a huge amount of noise.

Feature selection: Feature selection attempts to discover the attributes of a dataset that are most relevant to the data mining task at hand. It is a commonly used and powerful technique for reducing the dimensionality of a

problem to more manageable levels. Feature selection involves searching through various feature subsets and evaluating each of these subsets using some criterion (Guyon and Elisseeff, 2003). The evaluation criteria follow one of two basic models, the wrapper model (Dunne *et al*., 2002) and the filter model. The most popular search strategies are greedy sequential searches through the feature space, either forward or backward. Dhillon *et al*. (2003) proposed to combine global feature selection and model-based clustering. A recent approach on feature selection and model-based clustering is given by Raftery and Dean (2006). Frequently used filter methods include t-test (Hua *et al*., 2008), chi-square test (Jin *et al*., 2006), Wilcox on Mann-Whitney test (Liao *et al*., 2007), mutual information (Peng *et al*., 2005), Pearson correlation coefficients (Biesiada and Duch, 2008) and principal component analysis. Zhu *et al*. (2010) conducted a survey on feature selection.

Algorithms based on the filter model examine intrinsic properties of the data to evaluate the feature subset prior to data mining. Much effort in feature selection has been directed at supervised learning. Feature selection methods (Das and Liu, 2000) for supervised learning rely on the evaluation criteria like accuracy and/or on class labels. As we already know, in the unsupervised case we have neither class labels nor universally accepted evaluation criteria, but there are a number of methods that successfully adapt feature selection to clustering. However, while quite successful on a lot of datasets, feature selection algorithms have difficulty when clusters are found in different subspaces.

Feature clustering: We find early approaches of the feature clustering in text mining applications (Pereira *et al*., 1993), since this application domain suffers the curse of dimensionality very much. The feature clustering process is more effective than feature selection methods. At the same time, feature clustering avoids also one of the main drawbacks of the feature extraction methods: the amplification of noise due to the extraction of new features as combination of other features. In fact, in opposition to feature selection, feature clustering do not throw away any feature, but perform a clustering on the feature set and therefore it ideally keeps most relevant portion of information about all features. Furthermore, unlike feature extraction methods, feature clustering performs also a reduction of noise since a clustering algorithm can be also viewed as a compression algorithm. Dhillon *et al*. (2003), the authors achieve very good results exploiting an information theoretic framework to perform the feature clustering; their algorithm has good performance minimizing the intra-cluster divergence and simultaneously maximizing the inter-cluster divergence.

SUBSPACE BASED CLUSTERING

A subspace clustering problem is a search algorithm for interesting subsets of objects and their associated subsets of attributes. Since the first subspace clustering algorithm for data mining was proposed by Agrawal *et al*. (1998), many different algorithms have been presented by Yang *et al*. (2002), Zhou *et al*. (2007), Singh (2010) and Deng *et al*. (2010). All these algorithms can be classified into two categories: partition based approaches and grid based approaches (or density-based approaches).

Partition-based subspace clustering: Partition based algorithms partition the set of objects into mutually exclusive groups. Each group along with the subset of dimensions where this group of objects shows the greatest similarity is reported as a subspace cluster. Similar to the k-means method, most algorithms in this category define an objective function to be minimized during the search. The major difference between these methods and the k-means algorithm is that here the objective functions are related to the subspaces where each cluster resides in.

CLIQUE (Agrawal *et al*., 1998) was one of the first algorithms proposed that attempted to find clusters within subspaces of the dataset. As described above, the algorithm combines density and grid based clustering and uses an APRIORI style technique to find cluster able subspaces. Once the dense subspaces are found they are sorted by coverage where coverage is defined as the fraction of the dataset covered by the dense units in the subspace. 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. CLIQUE is able to find many types and shapes of clusters.

SUBCLU (density-connected SUBspace CLUstering) (Kailing *et al*., 2004) overcomes the limitations of grid-based approaches like the dependence on the positioning

of the grids. Instead of using grids the DBSCAN (Ester *et al*., 1996) cluster model of density-connected sets is used. SUBCLU is based on a bottom-up, greedy algorithm to detect the density-connected clusters in all subspaces of high-dimensional data. The algorithm starts with generating all 1-dimensional clusters w.r.t. the input parameters ϵ and μ by applying DBSCAN to each 1-dimensional subspace. Then, for each k-dimensional cluster it has to be checked iteratively if it is still existent in one ore more (k+1)-dimensional subspaces. For this purpose, all pairs of k-dimensional cluster having (k+1) attributes in common are joined together to generate (k+1)-dimensional candidate subspaces. In the last step of the iteration the (k+1)-dimensional clusters are generated by applying DBSCAN to each cluster of one k-dimensional subspace of each (k+1)-dimensional candidate subspace. These steps are recursively executed as long as the set of k-dimensional subspaces containing clusters is not empty. Compared to the grid-based approaches SUBCLU achieves a better clustering quality but requires a higher runtime.

PROCLUS (Aggarwal *et al*., 1999), a typical partition-based subspace clustering algorithm, searches for a partition of the dataset into clusters together with the set of dimensions on which each cluster is correlated. PROCLUS is a variation of the k-medoid algorithm and the number of clusters k and the average number of dimensions of clusters l need to be specified before the running of the algorithm. Furthermore, this algorithm assumes that each projected cluster has at least 2 dimensions. In the initialization stage, a set of random points are selected as the cluster medoids. In the iterative phase, data points that are close to each medoid are selected to determine the subspaces for the clusters and then each data point is assigned to its nearest cluster medoid. Normalized Manhattan segmented distance is used to measure the distance between data points in the context of subspaces. Quality of the current clusters is evaluated as the average Manhattan segmented distance from the points to the actual centroids of the clusters to which they are assigned to. A hill climbing technique is used to iteratively improve the quality of the clustering results until the termination criterion is met. This algorithm also assumes that one data point can be assigned to at most one subspace cluster or classified as an outlier, while a dimension can belong to multiple clusters.

ORCLUS (Aggarwal and Yu, 2000) is a generalization from PROCLUS (Aggarwal *et al*., 1999) which finds clusters in arbitrarily oriented subspaces. ORCLUS finds projected clusters as a set of data points C together with a set of orthogonal vectors such that these data points are closely clustered in the subspace defined by ^{2}. Similar to PROCLUS, the number of clusters k needs to be decided beforehand. Furthermore, ORCLUS requires that each cluster must have the same dimension l. Initially, k_0 (k_0>k) points are randomly selected as the cluster centroids and their vector spaces are set to be the original attribute space. In the iterative stage, each data point is assigned to its closest centroid measured by the projected Euclidean distance. Then the covariance matrix is computed for each cluster C_i and the set of eigenvectors corresponding to the q (q>l) smallest eigen values is selected as ^{2}i. Clusters near each other are merged, so the values of k_0 (number of clusters) and q (dimensionality of the subspaces) keep decreasing during the iterative phase. The iterative step stops when k_0 reduces to the predefined number of clusters k and q reduces to the predefined dimensionality of the subspaces l.

FIRES (Kriegel *et al*., 2005) is a general framework for efficient subspace clustering. It is generic in such a way that it works with all kinds of clustering notions. FIRES consist of the following three steps: pre clustering, generation of subspace cluster approximations and post processing. First, in the pre clustering step, all 1-dimensional clusters called base-clusters are computed. This is similar to existing subspace clustering approaches and can be done using any clustering algorithm of choice. In a second step, the base-clusters are merged to find maximal-dimensional subspace cluster approximations. However, they are note merged in an Apriori style but by using an algorithm that scales at most quadratic w.r.t. the number of dimensions. As a last step, a post processing step can be applied to refine the cluster approximations retrieved after the second step.

A CLICK (Zaki *et al*., 2007) uses a novel formulation of categorical subspace clusters, based on the notion of mining cliques in a k-partite graph. It implements an efficient algorithm to mine k-partite maximal cliques which correspond to the clusters. Using a novel vertical encoding we can guarantee the completeness of the results at a reasonable additional cost without sacrificing scalability. CLICKS imposes no domain constraint, is scalable to high dimensions, mines subspace and full-dimensional clusters and outperforms existing approaches by over an order of magnitude.

COSA (Friedman and Meulman, 2004) formalizes the subspace clustering problem as an optimization problem. The algorithm starts with all dimensions with equal weights for all data points. In the iterative phase, conventional full-dimensional clustering methods such as

the k-Nearest Neighbor algorithm are used to cluster the data points into clusters based on the current weights. Then the weights are re-computed to minimize an objective function based on the current clustering results. The algorithm keeps updating the clusters and the weights alternately until the weights stabilize. The motivation for this algorithm is that after several iterations, the weights for the dimensions of a subspace cluster become large for those data points that belong to this cluster. The most important parameter for this method controls the strength of incentive for clustering on more dimensions. The algorithm returns with mutually exclusive subspace clusters with each data point assigned to exactly one subspace cluster. One dimension can belong to more than one subspace clusters. However, the subspace in which each cluster is embedded is not explicitly known from the algorithm.

The FINDIT (Woo *et al*., 2004) algorithm which uses a dimension voting technique to find subspace clusters. Dimension oriented distance is defined to measure the distance between points based on not only the value information, but also the dimension information. Two points are considered similar in one dimension if their values in this dimension differ by less than $\in$ and the distance between them is the number of dimensions in which their differences are larger than $\in$. A smaller sample containing M data points is chosen randomly as the cluster medoids and a larger sample of size S is chosen as the data sample. For each medoid p$\in$M, the subspaces for the cluster is voted on by the nearest neighbors of p in S. Clusters that are too near to each other are merged to form larger clusters. The quality of a cluster is measured by the product of the number of data points contained in it and the dimensionality of the subspace of this cluster and it assumes that there is no overlap between clusters.

CLTree (Liu *et al*., 2000) finds subspace clusters in hyper-rectangular regions using the supervised decision tree building algorithm. All real data points are labeled as class `Y' and non-existing points are added on the y randomly during the running of the algorithm with the class label `N'. A decision tree is built to discriminate these two classes of data points and the regions containing mostly `Y' points are reported as the final subspace clusters. Several modifications have been made on the classic decision tree algorithm to accommodate the requirements of the subspace clustering problem. A user oriented final pruning is also proposed to refine the clustering results. Compared with other partition-based subspace clustering algorithms, CLTree has the advantage that it needs no prior input parameters. As a by-product, this algorithm returns not only the dense spots, but also the empty spots which may be useful in many application areas. But it still makes the assumption of no overlap between clusters.

Grid-based subspace clustering: Another view on the subspace clustering problem considers the data matrix as a high-dimensional grid and the clustering process as a search for dense regions in the grid. CLIQUE (Agrawal *et al*., 1998), the first subspace clustering algorithm for data mining applications, belongs to this category. Each dimension is partitioned into intervals of equal-length and an n-dimensional unit is the intersection of intervals from n distinct dimensions. A data point is contained in a unit if the values of all its attributes fall in the intervals of the unit for all dimensions of the unit. A unit is dense if the fraction of the total data points contained in it exceeds an input parameter. The algorithm starts the search for dense units from single dimensions. Candidate n-dimensional dense units are generated using the property that if a unit is dense in k dimensions, all its k_i1 dimensional projection units must all be dense. This downward closure property dramatically reduces the search space. Two units in the same subspace are connected if they share one common face or they are both connected to another unit. All connected dense units form clusters. Finally, clusters are defined to be hyper-rectangular maximal regions the union of that covers all the dense units. Each region is represented as a Disjunctive Norm Form (DNF) expression. A further pruning criterion is proposed to find only interesting subspaces using the coverage measurement. The coverage of a subspace is the ratio between the data contained in all dense units in this subspace and the total number of data points in the dataset. Based on the minimal description length technique, subspaces of low coverage are pruned during the iterative stage. However, this pruning may cause losing small clusters in those less dense subspaces. Since the number of candidate dense units grows exponentially in the highest dimensionality of the dense units, this algorithm becomes very inefficient when there are clusters in subspaces of high dimension.

ENCLUS (Cheng *et al*., 1999) uses entropy instead of density and coverage as a heuristic to prune away uninteresting subspaces. The algorithm shows that a subspace with clusters tends to have low entropy and under certain conditions, a subspace with high coverage also tends to have low entropy. Interest is defined as the difference between the sum of entropy of each individual dimension and the entropy of the multi-dimensional distribution. Subspaces with high interest indicate high

correlations between dimensions. The algorithm finds correlated, high density and high coverage subspaces using a similar level wise search algorithm to the one used in CLIQUE.

However, this algorithm finds only subspaces within which meaningful clusters exist, without explicitly finding the actual clusters.

pMAFIA (Nagesh *et al*., 2001) proposes to use adaptive units instead of the rigid ones used in CLIQUE. First, each dimension is partitioned into windows of small size and then adjacent windows having similar distribution are merged to form larger windows. Uniformly distributed dimensions are partitioned into a predefined number of equal-length windows. After all the 1-dimensional units are found, it uses a similar level wise search algorithm for dense units as with CLIQUE. The benefit of using the adaptive grids is that the number of candidate dense units in each iteration is reduced. Furthermore, since clusters are kept in their natural shape, there is no need to find connected dense units like CLIQUE. Parallelism is introduced to further speedup the algorithm. However, pMAFIA suffers from the same problem as CLIQUE, that is, the search complexity increases exponentially as a function of the highest dimensionality of the dense units.

DOC (Density-based Optimal Projective Clustering) (Procopiuc *et al*., 2002) proposes a mathematical definition of an optimal projective cluster" along with a Monte Carlo algorithm to compute approximations of such optimal projective clusters. A projective cluster is defined as a pair (C, D) where, C is a subset of the data set and D is a subset of the dimensions of the data space. Using the user specified input parameters ω and α an optimal projective cluster (C, D) is given if C contains more than α% of the data set and the projection of C into the subspace spanned by D must be contained in a hyper-cube of width ω whereas in all other dimensions $d \in D$ the points in C are not contained in a hyper-cube of width ω. Another parameter β has to be specified that defines the balance between the number of points in C and the number of dimensions in D. The proposed algorithm DOC only finds approximations because it generates projected clusters of width 2ω. In addition, no assumption on the distribution of points inside such a hyper-cube is made. The reported projected clusters may contain additional noise objects (especially when the size of the projected cluster is considerably smaller than 2ω) and or may miss some points that naturally belong to the projected cluster (especially when the size of the projected cluster is considerably larger than 2ω).

O-Cluster (Milenova and Campos, 2002), this clustering method combines a novel partitioning active sampling technique with an axis parallel strategy to identify continuous areas of high density in the input space. O-cluster is a method that builds upon the contracting projection concept introduced by opt grid. O-cluster makes two major contributions (1) it uses statistical test to validate the quality of a cutting plane. This statistical test identifies good splitting points along data projections. (2) It can operate on a small buffer containing a random sample from the original data set. Partitions that do not have ambiguities are "frozen" and the data points associated with them are removed from the active buffer. O-cluster operates recursively. It evaluates possible splitting points for all projections in a partition, selects the "best" one and splits the data into new partitions. The algorithm proceeds by searching for good cutting planes inside the newly created partitions. O-cluster creates a hierarchical tree structure that translates the input space into rectangular regions. The main processing stages are (1) load data buffer, (2) compute histograms for active partitions, (3) Find "best" splitting points for active partitions, (4) Flag ambiguous and "frozen" partitions, (5) Split active partitions and (6) Reload buffer. O-cluster is a non parametric algorithm. O-cluster functions optimally for large data sets with many records and high dimensionality.

EWKM (Jing *et al*., 2007) is a new k-means type subspace clustering algorithm for high-dimensional sparse data. This algorithm simultaneously minimize the within cluster dispersion and maximize the negative weight entropy in the clustering process. Because this clustering process awards more dimensions to make contributions to identification of each cluster, the problem of identifying clusters by few sparse dimensions can be avoided. As such, the sparsity problem of high-dimensional data is tackled. EWKM algorithm is outperformed other than k-means type algorithms and subspace clustering methods, for example, PROCLUS and COSA (Friedman and Meulman, 2004), in recovering clusters. Except for clustering accuracy, this algorithm is scalable to large high-dimensional data and easy to use because the input parameter is not sensitive. The weight values generated in the clustering process are also useful for other purposes, for instance, identifying the keywords to represent the semantics of text clustering results.

COMPARISONS AMONG SUBSPACE CLUSTERING ALGORITHMS

Here, we study the performance of various subspace clustering algorithms. We have used run time, arbitrary shape clustering and handle noise as parameters for evaluation of performance among subspace clustering algorithms. The results are presented in Table 1 and 2.

Table 1: Comparisons among subspace clustering algorithms

Algorithm	Data sets	Run time	Arbitrary shape clustering	Handle noise
CLIQUE	Real data sets: Insur1,Insur2,Store data (d-24), Bank data (d = 52),Synthetic data set (d-10 to 100 and n -10,00 to 100,000)	O (Ck+mk) K-Highest dimensionality, m-number of input points, C-number of clusters.	No	No
SUBCLU	Synthetic data set (d = 5 to 50 and n = 5000 to 30000), Gene expression data(n- 6000)	O (n log n)	Yes	Yes
PROCLUS	Ionosphere Data (d-34) Credit Report Data (d-2) Synthetic data set (d-5 to 40 and n-100,000)	O (k^3) k- No. of clusters	No	No
ORCLUS	Synthetic data set (n-5,000 to 100,000 and d-5 to 40)	O ($k_0^3+k_0$. N. d + k_0^2.d^3) k^0-initial number of seeds, d-dimensional space and N-no of points	No	No
FIRES	Bio1data (d- 4000), Bio2 data (d-7100 and n-72) Synthetic data set (n-1,000 and d- 50)	O (n)	No	Yes
CLICKS	Data 1 (d-10 and n =100), Data 2 (d-4 and n-one million), Mushroom data (n- 8124 and d- 22) and Congressional votes (n-435 and d-16)	O (n.m.$\|c_i\|$+2^l.$\|cl\|$+n.$\|c_i\|$.$\|f_i\|$) n-no of attributes, m-no of records of each attribute,$\|c_i\|$= Exponential number of cliques and $\|fl\|$ = No. of maximal csets.	No	Yes
COSA	Data 1(d -10000 and n -100), Data 2 (d -221 and n = 213) and Medical data set (d-11 and n-242)	O (hnmL + n^2m) h-no of iterations, k -no of clusters, n-number of objects, m-number of dimensionality and L-pre defined parameter to find L nearest neighbors objects.	Yes	No

Table 2: Comparisons among subspace clustering algorithms

Algorithm	Data sets	Run time	Arbitrary shape clustering	Handle noise
FINDIT	300 data sets (n-100,000 and d-20 to 50)	Ω ($\|S\|$ + $\|S\|$ log N/$\|S\|$) N-data set size and S-random sample size	No	Yes
CLTree	Synthetic data sets (d = 10 to 100 and n = 100,000 to 500,000)	O(n log n)	No	Yes
pMAFIA	Synthetic data sets(n = 1.4 million Real data sets: One Day Ahead Prediction of DAX (d-22 and n-2757) Ionosphere Data (n = 34 and d = 351)	O (ck'+N/pBk'γ+ αSpk')B-Number of records that fit in memory buffer, γ-I/O access time for a block of B records-Constant for communication, N-Total number of records, S-Size of messages exchanger among processors, P-Number of Processors and k'-Number of Dimensions.	Yes	No
ENCLUS	Synthetic data sets (n = 300,000 and d = 10 to 50)	O (ND+ m^D) N-no of transactions in database, D-total no of dimensionality and m-no of intervals in each dimension.	No	No
O-Cluster	Synthetic data sets(d-10 to 100 and n-50,000 to 400,000)	O (N×d) N-number of objects and d-number of dimensions.	Yes	Yes
DOC	Synthetic data sets (n = 5,000 to 100,000 and d = 5 to 40)	O ($nd^{5.322}$) n-no of data points and *d*-no of dimensions	No	No
EWKM	Text data sets-20-Newsgroups data and business transactions data.	O (hmnk) h-no of iterations, k-no of clusters, n-number of objects and m-number of dimensionality	Yes	Yes

DENSITY-BASED CLUSTERING ALGORITHMS

Density-based clustering methods group neighboring objects into clusters based on local density conditions rather than proximity between objects (Sun *et al*., 2008; Deng *et al*., 2010). These methods regard clusters as dense regions being separated by low density noisy regions. Density-based methods have noise tolerance and can discover no convex clusters. Similar to hierarchical and partitioning methods, density-based techniques encounter difficulties in high dimensional spaces because of the inherent sparsity of the feature space which in turn, reduces any clustering tendency. Some representative examples of density-based clustering algorithms are:

DBSCAN (Ester *et al*., 1996) seeks for core objects whose neighborhoods (radius) contains at least MinPts points. A set of core objects with overlapping neighborhoods define the skeleton of a cluster. Non-core points lying inside the neighborhood of core objects represent the boundaries of the clusters, while the remaining is noise. DBSCAN can discover arbitrary-shaped clusters, is insensitive to outliers and order of data input, while its complexity is O (N^2). If a spatial index data structure is used the complexity can be improved up to O (N log N). DBSCAN breaks down in high dimensional spaces and is very sensitive to the input parameters and MinPts.

OPTICS(Ordering Points To Identify the Clustering Structure) (Ankerst *et al*., 1999), an extension of DBSCAN

(Ester *et al*., 1996) to adapt to local densities, builds an augmented ordering of data and stores some additional distance information, allowing the extraction of all density-based clustering for any lower value of the radius. OPTICS has the same complexity as that of DBSCAN.

GCHL-A grid-clustering algorithm for high-dimensional very large spatial data bases (Pilevar and Sukumar, 2005) which groups similar spatial objects into classes, is an important component of spatial data mining. Due to its immense applications in various areas, spatial clustering has been highly active topic in data mining researches, with fruitful, scalable clustering methods developed recently. These spatial clustering methods can be classified into four categories: partitioning method, hierarchical method, density-based method and grid-based method. Clustering large data sets of high dimensionality has always been a serious challenge for clustering algorithms. Many recently developed clustering algorithms have attempted to address either handling data with very large number of records or data sets with very high number of dimensions. This new clustering method GCHL (a Grid-Clustering algorithm for High-dimensional very Large spatial databases) combines a novel density-grid based clustering with axis-parallel partitioning strategy to identify areas of high density in the input data space. The method operates on a limited memory buffer and requires at most a single scan through the data. GCHL demonstrate the high quality of the obtained clustering solutions, capability of discovering deeper and higher regions, their robustness to outlier and noise and GCHL excellent scalability.

DENCLUE (Hinneburg and Keim, 1998) uses an influence function to describe the impact of a point about its neighborhood while the overall density of the data space is the sum of influence functions from all data. Clusters are determined using density attractors, local maxima of the overall density function. To compute the sum of influence functions a grid structure is used. DENCLUE scales well, can find arbitrary-shaped clusters, is noise resistant, is insensitive to the data ordering, but suffers from its sensitivity to the input parameters. The curse of dimensionality phenomenon heavily affects DENCLUE's effectiveness. Moreover, similar to hierarchical and partitioning techniques, the output, e.g., labeled points with cluster identifier, of density-based methods can not be easily assimilated by humans. The DENCLUE (Hinneburg and Keim, 1998) algorithm employs a cluster model based on kernel density estimation. A cluster is defined by a local maximum of the estimated density function. Data points are assigned to clusters by hill climbing, i.e., points going to the same local maximum are put into the same cluster. A disadvantage of DENCLUE 1.0 (Hinneburg and Keim, 2003) is that the used hill climbing may make unnecessary small steps in the beginning and never converges exactly to the maximum, it just comes close then DENCLUE2.0 (Hinneburg and Gabriel, 2007) was introduced, this new hill climbing procedure for Gaussian kernels adjusts the step size automatically at no extra costs. This procedure converges exactly towards a local maximum by reducing it to a special case of the expectation maximization algorithm. The new procedure needs much less iterations and can be accelerated by sampling based methods with sacrificing only a small amount of accuracy.

Rough-DBSCAN (Viswanath and Suresh Babu, 2009) is a Density based clustering techniques like DBSCAN are attractive because it can find arbitrary shaped clusters along with noisy outliers. Its time requirement is O (n^2) where n is the size of the dataset and because of this it is not a suitable one to work with large datasets. This algorithm apply the leaders clustering method first to derive the prototypes called leaders from the dataset which along with prototypes preserves the density information also, then to use these leaders to derive the density based clusters. Rough-DBSCAN (Viswanath and Suresh Babu, 2009) has a time complexity of O (n) only and is analyzed using rough set theory. The authors shown that for large datasets rough-DBSCAN can find a similar clustering as found by the DBSCAN (Ester *et al*., 1996) but are consistently faster than DBSCAN.

In DENCOS (Chu *et al*., 2010) different density thresholds is utilized to discover the clusters in different subspace cardinalities to cop up with density divergence problem. Here the dense unit discovery is performed by utilizing a novel data structure DFP-tree (Density FP-tree) which is constructed on the data set to store the complete information of the dense units. From the DFPtree, it computes the lower bounds and upper bounds of the unit counts for accelerating the dense unit discovery and this information are utilized in a divide-and-conquer scheme to mine the dense units. Therefore, DENCOS is devised as a two-phase algorithm comprised of the preprocessing phase and the discovering phase. The preprocessing phase is to construct the DFP-tree on the transformed data set where the data set is transformed with the purpose of transforming the density conscious subspace clustering problem into a similar frequent item set mining problem. Then, in the discovering phase, the DFP-tree is employed to discover the dense units by using a divide-and-conquer scheme. The Generalized Grid File (GGF), using a parameter that defines the number of attributes for which an access is noted as GGF (1). If single-attribute access has to be supported, then GGF behaves like a B+-tree and like a grid file if the number of attributes is greater

than one. However, the identification of dense regions in previous works lacks of considering a critical problem, called "the density divergence problem" in this algorithm which refers to the phenomenon that the region densities vary in different subspace cardinalities. Without considering this problem, previous works utilize a density threshold to discover the dense regions in all subspaces which incurs the serious loss of clustering accuracy (either recall or precision of the resulting clusters) in different subspace cardinalities. To tackle the density divergence problem, this algorithm devise a novel subspace clustering model to discover the clusters based on the relative region densities in the subspaces where the clusters are regarded as regions whose densities are relatively high as compared to the region densities in a subspace. Based on this idea, different density thresholds are adaptively determined to discover the clusters in different subspace cardinalities. As validated by our extensive experiments on various data sets, DENCOS can discover the clusters in all subspaces with high quality and the efficiency of DENCOS outperforms previous works.

MITOSIS (Yousria *et al*., 2009) finds arbitrary shapes of arbitrary densities in high dimensional data. Unlike previous algorithms, this algorithm uses a dynamic model that combines both local and global distance measures. The algorithm's ability to distinguish arbitrary densities in a complexity of order O ($D_n \log_2 (n)$) renders it attractive to use. Its speed is comparable to simpler but less efficient algorithms and its efficiency is comparable to efficient but computationally expensive ones. Its ability to distinguish outliers is also of agree at importance in high dimensions. Moreover, introducing an accompanying parameter selection procedure makes Mitosis more convenient to use, compared to related algorithms. The experimental results illustrate the efficiency of Mitosis, compared to ground truth, for discovering relatively low and high density clusters of arbitrary shapes. The use of real high dimensional data sets supports its applicability in real life applications. Validity indexes indicate that Mitosis out performs related algorithms as DBSCAN (Ester *et al*., 1996) which finds clusters of arbitrary shapes. It is also compared to a center-based algorithm, illustrating the importance of discovering natural cluster shapes.

V3COCA(Wang *et al*., 2009) an effective clustering algorithm for complicated objects and its application in breast cancer research and diagnosi which can resolve several issues that have not or only partially, been addressed by existing clustering algorithms. This algorithm allows the users to use the algorithm without providing any parameter input, or to use it with a series of objects as input. Unlike most of the input dependent algorithms, the V3COCA algorithm calculates the necessary parameters automatically. It generates explicit clusters to the users for computer aided diagnosis and disease research. It recognizes noises from breast cancer objects. As some of the objects contain errors or have part of the information missed, these objects are not expected to be categorized into any clusters. It creates arbitrary shaped clusters, with different densities. It adopts a new distance definition to describe the dissimilarity of two breast cancer objects. Traditional distance definitions could not be used because the features composing a breast cancer object may be numerical or nominal, or have different medical importance. The new definition makes reasonable transformation from nominal value to numerical value and gives different flexible weight values to different features. The V3COCA meets all the requirements. Although is more powerful, the time complexity of V3COCA is not higher than the existing algorithms. It took several seconds longer in execution time than DBSCAN (Ester *et al*., 1996) and K-means in certain cases, but V3COCA is always faster than the Hierarchical and OPTICS (Ankerst *et al*., 1999).Overall, the V3COCA generates far more satisfying clustering results within an acceptable level of execution time.

PACA-DBSCAN (Jiang *et al*., 2011) is based on partitioning-based DBSCAN and modified ant clustering algorithms. It can partition database into N partitions according to the density of data, then cluster each partition with DBSCAN (Ester *et al*., 1996). Superior to DBSCAN, The new hybrid algorithm reduces the sensitivity to the initial parameters and can deal with data of uneven density very well. For multi-dimensional data, the PACA-DBSCAN algorithm does not need to discuss the distribution of data on each dimension. In contrast with DBSCAN, The PACA-DBSCAN can correctly cluster data of very special shape. The results of PACA-DBSCAN are evaluated and compared by the classical F-Measure and a proposed criterion (ER). The algorithm has proved that the performance of PACA-DBSCAN is better than DBSCAN.

APSCAN (Chen *et al*., 2011) is a parameter free clustering algorithm. Firstly, it utilizes the Affinity Propagation (AP) algorithm to detect local densities for a dataset and generate a normalized density list. Secondly, it combines the first pair of density parameters with any other pair of density parameters in the normalized density list as input parameters for a proposed DDBSCAN (Double-Density-Based SCAN) to produce a set of clustering results. In this way, it can obtain different clustering results with varying density parameters derived from the normalized density list. Thirdly, it develops an

Table 3: Comparisons among of density based clustering algorithms

Algorithm	Data sets	Run time	Arbitrary shape clustering	Handle noise
DBSCAN	SEQUOIA 2000 benchmark	O (n log n)	Yes	Yes
ROUGH-DBSCAN	banana dataset (d-2),Pen digits dataset (d-30 and n = 1000) and Shuttle dataset (n = 5000)	O (n)	Yes	Yes
OPTICS	Color histograms (n = 10,000 to 100,000 and d = 64)	O (n log n)	Yes	Yes
GCHL	DS1(n-2000 and d = 5) and DS2 (n=100,000 and d = 23)	O (N.ρ.d) to O (ρ.d.N. log N) N = no of blocks, D = no of dimensions and ρ-no of blocks)	No	Yes
DENCLUE	Synthetic data set (d = 11 and n = 20000 to 100000)	O(n log n) n-no of data points	Yes	Yes
DENCOS	Synthetic data sets: DS1 (d = 10 and n = 13965) ,DS2 (d = 10 and n = 30580), DS3 (d= 10 and n = 4420)and adult database (n = 32561 and d = 6) and thyroid disease (n = 18152 and d = 6)	$O(d^k)$ d-dimensionality of data set and k-cardinality of data set	Yes	Yes
MITOSIS	Synthetic control charts data set (n-600 and d-60) , Cylinder bell funnel(n-900 and d-128), Optical character recognition. (n = 5620 and d = 64) and Pen digit character (n = 10992 and d = 16)	O (n log n)	Yes	Yes
V3COCA	10,000 breast cancer cases	O (n log n)	Yes	Yes
PACA-DBSCAN	Artset1(d-3 and n-300), Artset2 (d-4 and n-1572), Iris (d-3 and n = 150) and wine (d-3 and n-178)	O (n)	Yes	Yes
APSCAN	Toy dataset 1 (n = 1000) and Toy dataset 2 (n = 500)	O (n)	Yes	Yes

updated rule for the results obtained by implementing the DDBSCAN with different input parameters and then synthesizes these clustering results into a final result. The APSCAN has two advantages: first it does not need to predefine the two parameters as required in DBSCAN (Ester *et al*., 1996) and second, it not only can cluster datasets with varying densities but also preserve the nonlinear data structure for such datasets.

COMPARISON AMONG DENSITY BASED CLUSTERING ALGORITHMS

In this section, we study the performance of various density based clustering algorithms. We have used run time, arbitrary shape clustering and handle noise as parameters for evaluation of performance among these algorithms. Details of performance observations among density based clustering algorithms are given in Table 3.

CONCLUSION

Clustering techniques have been studied extensively in the areas of statistics, machine learning and database communities in the past decades. In most clustering approaches, the data points in a given data set are partitioned into clusters such that the points within a cluster are more similar among themselves than data points in other clusters. However, traditional clustering techniques fall short when clustering is performed in high dimensional spaces. In order to overcome the limitations of traditional clustering algorithms, some attempts have been made to use genetic algorithms for clustering data sets.

As the number of dimensions increase, many clustering techniques begin to suffer from the curse of dimensionality, degrading the quality of the results. Densities also suffer from the curse of dimensionality. Density-based clustering methods hence have problems to determine the density of a region as the objects are scattered over the data space.

There are two traditional ways to tackle the problem of high dimensionality. The first one consists in a variety of techniques to perform a dimensionality reduction before the clustering process start. The second way is known as subspace clustering and density based clustering.

Subspace clustering has been proposed to overcome this challenge and has been studied extensively in recent years. The goal of subspace clustering is to locate clustering in different subspaces of the same data set. In general, a subspace cluster represents not only the cluster it self, but also the subspace where the cluster is situated.

The two main categories of subspace clustering algorithms are partition based approaches and grid based approaches. First, partition based algorithms partition the set of objects into mutually exclusive groups. Each group along with the subset of dimensions where this group of objects shows the greatest similarity is reported as a subspace cluster. Second, the grid based subspace clustering algorithms consider the data matrix as a high-dimensional grid and the clustering process as a search for dense regions in the grid. Density-based clustering methods group neighboring objects into clusters based on local density conditions rather than proximity between objects. These methods regard clusters as dense regions being separated by low density noisy

regions. Density-based methods have noise tolerance and can discover no convex clusters.

In this study, we have discussed clustering problem with high dimensional data and approaches to solve this problem. Specifically, we discussed subspace based clustering approaches and density based approaches. We presented the latest developments in this area. This study will also provide direction to the researchers who would like explore more effective approaches for clustering of high dimensional data.

REFERENCES

Agrawal, R., J. Gehrke, D. Gunopulos and P. Raghavan, 1998. Automatic subspace clustering of high dimensional data for data mining applications. Proceedings of ACM SIGMOD International Conference on Management of Data, June 1-4, ACM Press, New York, pp: 94-105.

Aggarwal, C.C., C. Procopiuc, J.L. Wolf, P.S. Yu and J.S. Park, 1999. Fast algorithms for projected clustering. Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data (ICMD'99), Philadelphia, PA., USA., pp: 1-12.

Aggarwal, C.C. and P.S. Yu, 2000. Finding generalized projected clusters in high dimensional spaces. Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data (ICMD'00), ACM Press, USA., pp: 70-81.

Amir, A. and D. Lipika, 2007. A k-mean clustering algorithm for mixed numeric and categorical data. Data Knowledge Eng., 63: 503-527.

Ankerst, M., M.M. Breunig, H.P. Kriegel and J. Sander, 1999. OPTICS: Ordering points to identify the clustering structure. Proceedings of ACM SIGMOD International Conference on Management of Data, May 31-June 3, Philadelphia, PA., pp: 49-60.

Arora, A., S. Upadhyaya and R. Jain, 2009. Integrated approach of reduct and clustering for mining patterns from clusters. Inform. Technol. J., 8: 173-180.

Bandyopadhyay S. and U. Maulik, 2002. An evolutionary technique based on K-means algorithm for optimal clustering in RN. Inform. Sci., 146: 221-237.

Bandyopadhyay, S. and S. Saha, 2007. A clustering method using a new point symmetry-based distance measure. Pattern Reconition., 40: 3430-3451.

Beyer, K., J. Goldstein, R. Ramakrishna and U. Shaft, 1999. When is nearest neighbors meaningful. Proceedings of the 7th International Conference on Database Theory (ICDT'99), London, UK., pp: 217-235.

Biesiada, J. and W. Duch, 2008. Feature selection for high-dimensional data-a Pearson redundancy based filter. Adv. Soft Comput., 45: 242-249.

Bishop, C.M., 2006. Pattern Recognition and Machine Learning. 1st Edn., Springer, Heidelberg, ISBN: 0-387-31073-8.

Chen, X., W. Liu, H. Qiu and J. Lai, 2011. APSCAN: A parameter free algorithm for clustering. Pattern Recognit. Lett., 32: 973-986.

Cheng, C.H., A. Waichee and F.Y. Zhang, 1999. Entropy-based subspace clustering for mining numerical data. Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Aug. 15-18, San Diego, California, United States, pp: 84-93.

Chu, Y.H., Y.J. Chen, D.H. Yang and M.S. Chen, 2009. Reducing redundancy in subspace clustering. IEEE Trans. Knowledge Data Eng., 21: 1432-1446.

Chu, Y.H., J.W. Huang, K.T. Chuang, D.N. Yang and M.S. Chen, 2010. Density conscious subspace clustering for high-dimensional data. IEEE Trans. Knowledge Data Eng., 22: 16-30.

Cutting, D., J. Carger, J. Pedersen and J. Tukey, 1992. Scatter/gather: A cluster-based approach to browsing large document collections. Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, June 21-24, Copenhagen, Denmark, pp: 318-329.

Das, M. and H. Liu, 2000. Feature selection for clustering. Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining (PADKK'00), Springer-Verlag, London, UK., pp: 110-121.

Defays, D., 1977. An efficient algorithm for a complete link method. Comput. J., 20: 364-366.

Deng, Z., K.S. Choi, F.L. Chung and S. Wang, 2010. Enhanced soft subspace clustering integrating with in-cluster and between-cluster information. Pattern Recognit., 43: 767-781.

Dhillon, I.S., S. Mallela and R. Kumar, 2003. A divisive information-theoretic feature clustering algorithm for text classification. J. Mach. Learn., 3: 1265-1287.

Dunne, K., P. Cunningham and F. Azuaje, 2002. Solutions to instability problems with sequential wrapper-based approaches to features selection. Dublin, Trinity College Dublin, Department of Computer Science, TCD-CS-2002-28, pp: 22. http://www.tara.tcd.ie/handle/2262/13144.

Ester, M., H.P. Kriegel, J. Sander and X. Xu, 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (ICKDDM'96), Portland, pp: 226-231.

Frank, I.E. and T. Roberto, 1994. Data Analysis Handbook. Elsevier Science Inc., New York, pp: 227-228.

Friedman, J.H. and J.J. Meulman, 2004. Clustering objects on subsets of attributes. J.R. Stat. Soc. Ser. B, 66: 815-849.

Gao, J., P.W. Kwan and Y. Guo, 2009. Robust multivariate L1 principal component analysis and dimensionality reduction. Neurocomputing, 72: 1242-1249.

Garg, S. and R.C. Jain, 2006. Variations of K-mean algorithm: A study for high-dimensional large data sets. Inform. Technol. J., 5: 1132-1135.

Girard, S., 2000. A nonlinear PCA based on manifold approximation. Comput. Stat., 15: 145-167.

Guyon, I. and A. Elisseeff, 2003. An introduction to variable and feature selection. J. Mach. Learn. Res., 3: 1157-1182.

Han, J. and M. Kamber, 2001. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, San Francisco, CA..

Hartigan, J.A., 1975. Clustering Algorithms. John Wiley and Sons, New York.

Hinneburg, A. and D.A. Keim, 1998. An efficient approach to clustering in large multimedia databases with noise. Proceedings of 4rth International Conference on Knowledge Discovery and Data Mining, Aug. 27-31, New York, pp: 58-65.

Hinneburg, A. and D.A. Keim, 2003. A general approach to clustering in large databases with noise. Knowledge Inf. Syst., 5: 387-415.

Hinneburg, A. and H.H. Gabriel, 2007. DENCLUE 2.0: Fast clustering based on kernel density estimation. Adv. Intell. Data Anal., 4723: 70-80.

Hua, J., W. Tembe and E.R. Dougherty, 2008. Feature selection in the classification of high-dimension data. Proceedings of the IEEE International Workshop on Genomic Signal Processing and Statistics, June 8-10, Phoenix, AZ., USA., pp: 1-2.

Jain, A.K., M.N. Murty and P.J. Flynn, 1999. Data clustering: A review. ACM Comput. Surveys, 31: 264-323.

Jiang, H., J. Li, S. Yi, X. Wang and X. Hu, 2011. A new hybrid method based on partitioning-based DBSCAN and ant clustering. Expert Syst. Appl. (In Press). 10.1016/j.eswa.2011.01.135

Jin, X., A. Xu, R. Bie and P. Guo, 2006. Machine learning techniques and chi-square feature selection for cancer classification using SAGE gene expression profiles. Lect. Notes Comput. Sci., 3916: 106-115.

Jing, L., M.K. Ng and J.Z. Huang, 2007. An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data. IEEE Trans. Knowledge Data Eng., 19: 1026-1041.

Jolliffe, I.T., 2002. Principal Component Analysis. 2nd Edn., Springer-Verlag, New York.

Kailing, K., H.P. Kriegel and P. Kroger, 2004. Density-connected subspace clustering for high-dimensional data. Proceedings of the 4th SIAM International Conference on Data Mining (SDM'04), Lake Buena Vista, FL., pp: 246-257.

Kaufman, L. and P.J. Rousseeuw, 1990. Finding Groups in Data: An Introduction to Cluster Analysis. Wiley, New York..

Kolatch, E., 2001. Clustering algorithms for spatial data bases: A survey. Department of Computer Science, University of Maryland, College Park, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.28.1145.

Kriegel, H.P., P. Kroger, M. Renz and S. Wurst, 2005. A generic framework for efficient subspace clustering of high-dimensional data. Proceedings of the 5th IEEE International Conference Data Mining, Nov. 27-30, Institute for Computer Science, Munich University, Germany, pp: 8-8.

Liao, C. and S. Li and Z. Luo, 2007. Gene selection using Wilcox on rank sum test and support vector machine for cancer. Lect. Notes. Comput. Sci., 4456: 57-66.

Liu, B., Y. Xia and P.S. Yu, 2000. Clustering through decision tree construction. Proceedings of the 9th International Conference on Information and Knowledge Management (CIKM'00), ACM Press, USA., pp: 20-29.

Macqueen, J.B., 1967. Some methods for classification and analysis of multivariate observations. Proc. Berkeley Symp. Math. Statist. Prob., 1: 281-297.

Milenova, B.L. and M.M. Campos, 2002. O-cluster: Scalable clustering of large high dimensional data sets. Proceedings of the IEEE International Conference on Data Mining (ICDM'02), New York, USA., pp: 290-297.

Nagesh, H., S. Goil and A. Choudhary, 2001. Adaptive grids for clustering massive data sets. Proceedings of the 1st SIAM International Conference on Data Mining (SDM'01), New York, USA., pp: 1-17.

Parsons, L., E. Haque and H. Liu, 2004. Subspace clustering for high dimensional data: A review. ACM SIGKDD Explorat. Newslett., 6: 90-105.

Peng, H., F. Long and C. Ding, 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance and min redundancy. IEEE Tran. Pattern Anal. Mach. Intell., 27: 1226-1238.

Pereira, F., N. Tishby and L. Lee, 1993. Distributional clustering of English words. Proceedings of the 31st annual Meeting on Association for Computational Linguistics (ACL'93), Stroudsburg, PA, USA., pp: 183-190.

Pilevar, A.H. and M. Sukumar, 2005. GCHL: A grid-clustering algorithm for high-dimensional very large spatial data bases. Pattern Recogn. Lett., 26: 999-1010.

Procopiuc, C.M., M.J. Pankaj, K. Agarwal and T.M. Murali, 2002. A monte carlo algorithm for fast projective clustering. Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, Jun. 03-06, Madison, Wisconsin, pp: 418-427.

Raftery, A.E. and N. Dean, 2006. Variable selection for model-based clustering. J. Am. Statist. Assoc., 101: 168-178.

Ranjan, J. and S. Khalil, 2007. Clustering methods for statistical analysis of genome databases. Inform. Technol. J., 6: 1217-1223.

Roweis, S.T. and L.K. Saul, 2000. Nonlinear dimensionality reduction by locally linear embedding. Science, 290: 2323-2326.

Sammon, Jr. J.W., 1969. A nonlinear mapping for data structure analysis. Trans. Comput., 18: 401-409.

Scholkopf, B., A.J. Smola and K.R. Muller, 1998. Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput., 10: 1299-1319.

Schott, J.R., 1993. Dimensionality reduction in quadratic discriminate analysis. Comput. Statist. Data Anal., 66: 161-174.

Sibson, R., 1973. SLINK: An optimally efficient algorithm for the single-link cluster method. Comput. J., 16: 30-34.

Singh, V., 2010. Mining subspaces from high dimensional data. Proceedings of National Conference on ICT: Theory Application and Practices, March 5-6, SPSU University Press, Udaipur, India, pp: 101-103.

Singh, V., L. Sahoo and A. Kelkar, 2010. Mining subspace clusters in high dimensional data. Int. J. Recent Trends Eng. Technol., 3: 118-122.

Singh, V. and P. Trikha, 2011. Density based algorithm with automatic parameters generation. Proceeding of the IEEE 3rd International Conference on Machine Learning and Computing, Feb. 26-28, Singapore, pp: 555-558.

Singh, V., L. Sahoo and A. Kelkar, 2011. Mining clusters in data sets of data mining: An effective algorithm. Int. J. Comput. Ther. Eng., 3: 171-177.

Sun, J.G., J. Liu and L.Y. Zhao, 2008. Clustering algorithms research. J. Software, 19: 48-61.

Tseng, L.Y. and S.B. Yang, 2001. A genetic approach to the automatic clustering problem. Pattern Recognit., 34: 415-424.

Velmurugan, T. and T. Santhanam, 2011. A survey of partition based clustering algorithms in data mining: An experimental approach. Inform. Technol. J., 10: 478-484.

Vijendra, S., L. Sahoo and K. Ashwini, 2010. An effective clustering algorithm for data mining. Proceedings of the International Conference on Data Storage and Data Engineering, Feb. 9-10, Bangalore, India, pp: 250-253.

Viswanath, P. and V. Suresh Babu, 2009. Rough-DBSCAN: A fast hybrid density based clustering method for large data sets. Pattern Recognit. Lett., 30: 1477-1488.

Wang, K., Z. Du, Y. Chen and S. Li, 2009. V3COCA: An effective clustering algorithm for complicated objects and its application in breast cancer research and diagnosis. Simul. Modell. Pract. Theory, 17: 454-470.

Woo, K.G., J.H. Lee, M.H. Kim and Y.J. Lee, 2004. FINDIT: A fast and intelligent subspace clustering algorithm using dimension voting. Inf. Software Technol., 46: 255-271.

Xu, R. and D. Wunsch II, 2005. Survey of clustering algorithms. IEEE Trans. Neural Networks, 16: 645-678.

Yang, J., W. Wang, H. Wang and P. Yu, 2002. δ-clusters: Capturing subspace correlation in a large data set. Proceedings of the 18th International Conference on Data Engineering, Feb. 26-March 01, San Jose, CA., USA., pp: 517-528.

Yousria, N.A., M.S. Kamel and M.A. Ismail, 2009. A distance-relatedness dynamic model for clustering high dimensional data of arbitrary shapes and densities. Pattern Recognit., 42: 1193-1209.

Zaki, M.J., M. Peters, I. Assent and T. Seidl, 2007. CLICKS: An effective algorithm for mining subspace clusters in categorical datasets. Data Knowledge Eng., 60: 51-70.

Zhou, H., B. Feng, L. Lv and Y. Hui, 2007. A robust algorithm for subspace clustering of high-dimensional data*. Inform. Technol. J., 6: 255-258.

Zhu, Z., Y.F. Guo, X. Zhu and X. Xue, 2010. Normalized dimensionality reduction using nonnegative matrix factorization. Neuro Comput., 73: 1783-1793.