Subspace Clustering for High Dimensional Categorical Data

Guojun Gan
Department of Mathematics and Statistics
York University
Toronto, Canada
gjgan@mathstat.yorku.ca

Jianhong Wu
Department of Mathematics and Statistics
York University
Toronto, Canada
wujh@mathstat.yorku.ca

ABSTRACT

Data clustering has been discussed extensively, but almost all known conventional clustering algorithms tend to break down in high dimensional spaces because of the inherent sparsity of the data points. Existing subspace clustering algorithms for handling high-dimensional data focus on numerical dimensions. In this paper, we designed an iterative algorithm called SUBCAD for clustering high dimensional categorical data sets, based on the minimization of an objective function for clustering. We deduced some cluster memberships changing rules using the objective function. We also designed an objective function to determine the subspace associated with each cluster. We proved various properties of this objective function that are essential for us to design a fast algorithm to find the subspace associated with each cluster. Finally, we carried out some experiments to show the effectiveness of the proposed method and the algorithm.

General Terms

Subspace Clustering

Keywords

Clustering, Subspace Clustering, Categorical Data

1. INTRODUCTION

Clustering has been used extensively as a primary tool of data mining. Many clustering algorithms have been designed [15; 14]. Unfortunately, most of these conventional clustering algorithms do not scale well to cluster high dimensional data sets in terms of effectiveness and efficiency, because of the inherent sparsity of high dimensional data. In high dimensional data sets, we encounter several problems. First of all, the distance between any two data points becomes almost the same [5], therefore it is difficult to differentiate similar data points from dissimilar ones. Secondly, clusters are embedded in the subspaces of the high dimensional data space, and different clusters may exist in different subspaces of different dimensions [3]. Because of these problems, almost all conventional clustering algorithms fail to work well for high dimensional data sets. One possible solution is to use dimension reduction techniques such as PCA(Principal Component Analysis) [27] and Karhunen-Loève Transformation [3], or feature selection techniques.

In dimension reduction approaches, one first reduces the dimensionality of the original data set by removing less important variables or by transforming the original data set into one in a low dimensional space, and then applies conventional clustering algorithms to the new data set. In feature selection approaches, one finds the dimensions on which data points are correlated. In either dimension reduction approaches or feature selection approaches, it is necessary to prune off some variables, which may lead to a significant loss of information. This can be illustrated by considering a 3-dimension data set that has 3 clusters: one is embedded in (x, y)-plane, another is embedded in (y, z)-plane and the third one is embedded in (z, x)-plane. For such a data set, an application of a dimension reduction or a feature selection method is unable to recover all the clustering structures, because the 3 clusters are formed in different subspaces. In general, clustering algorithms based on dimension reduction or feature selection techniques generate clusters that may not fully reflect the original clusters' structure.

This difficulty that conventional clustering algorithms encounter in dealing with high dimensional data sets motivates the concept of subspace clustering or projected clustering[3] whose goal is to find clusters embedded in subspaces of the original data space with their own associated dimensions.

Almost all the subspace clustering algorithms proposed so far are designed for clustering high dimensional numerical data sets. In this paper, we present SUBCAD(SUBspace clustering for high dimensional CAtegorical Data), a subspace clustering algorithm for clustering high dimensional categorical data sets. We shall develop a method to determine the subspace associated with each cluster, and we shall design an iterative method to cluster high dimensional categorical data sets by treating the clustering process as an optimization problem.

2. RELATED WORK

Since a subspace clustering algorithm CLIQUE [3] was first proposed by Aggarwal et. al., several subspace clustering algorithms have been designed[3; 9; 11; 19; 1; 2; 25; 8; 22; 17]. The recent subspace clustering algorithms can be roughly divided into three categories: Grid-based algorithms such as CLIQUE [3], MAFIA [11; 19], Partitioning and/or hierarchical algorithms such as ORCLUS [2],FINDIT [25], and Neural Network-based algorithms such as PART [8](See Table 1).

AT	Algorithms	DT	H/P
Grid-based	CLIQUE [3]	Num	O
	ENCLUS [9]	Num	O
	MAFIA [11; 19]	Num	O
Partitioning	PROCLUS [1]	Num	P
	ORCLUS [2]	Num	P
	FINDIT [25]	Num	P
	FLOC [26]	Num	P
	DOC [22]	Num	P
Neural Network	PART [8]	Num	H
Other	CLTree[17]	Num	O

Table 1: A list of some subspace clustering algorithms. Data Type(**DT**) indicates the type of data sets which the algorithm can be applied to, **AT** refers to Algorithm Type, Num refers to Numerical, H/P refers to Hierarchical/Partitioning, O refers to Other.

In the algorithm of CLIQUE, it first partitions the whole data space into non-overlapping rectangular units, and then searches for dense units and merges them to form clusters. The subspace clustering is achieved due to the fact that if a k-dimension unit $(a_1, b_1) \times (a_2, b_2) \times \cdots \times (a_k, b_k)$ is dense, then any $k-1$-dimension unit $(a_{i_1}, b_{i_1}) \times (a_{i_2}, b_{i_2}) \times \cdots \times (a_{i_{k-1}}, b_{i_{k-1}})$ is also dense, where (a_i, b_i) is the interval of the unit in the i-th dimension, $1 \leq i_1 < i_2 < \cdots < i_{k-1} \leq k$. ENCLUS(Entropy-based CLUStering) [9] and MAFIA(Merging of Adaptive Finite Intervals) [19; 11] are also Grid-based subspace clustering algorithms.

PROCLUS [1] is a variation of k-Medoid algorithm [15] for subspace clustering. The PROCLUS algorithm finds out the subspace dimensions of each cluster via a process of evaluating the locality of the space near it. FINDIT(a Fast and INtelligent subspace clustering algorithm using DImension voTing)[25], ORCLUS(arbitrarily ORiented projected CLUSter generation) [2], FLOC [26] and DOC(Density-based Optimal projective Clustering) [22] are also partitioning subspace clustering algorithms.

PART [8](Projective Adaptive Resonance Theory) is a new neural network architecture that was proposed to find projected clusters for data sets in high dimensional spaces. In PART, a so-called selective output signaling mechanism is provided in order to deal with the inherent sparsity in the full space of the high dimensional data points. PART is very effective to find the subspace in which a cluster is embedded, but the difficulty of tuning some parameters in the algorithm of PART restricts its application. CLTree(CLustering based on decision Trees)[17] is an algorithm for clustering numerical data based on a supervised learning technique called decision tree construction. The resulting clusters found by CLTree are described in terms of hyper-rectangle regions. The CLTree algorithm is able to separate outliers from real clusters effectively, since it naturally identifies sparse and dense regions.

3. ITERATIVE METHODS

To describe the algorithm, we start with some notations. Given a data set D, let Q be the set of dimensions of D, i.e. $Q = \{1, 2, ..., d\}$, where d is the number of dimensions of D, let Span(Q) denote the full space of the data set, then by a subspace cluster, we mean a cluster C associated with a set of dimensions P such that **(a)** The data points in C are "similar" to each other in the subspace Span(P) of Span(Q)(i.e. the data points in C are compact in this subspace); **(b)** The data points in C are sparse in the subspace Span(R), where $R = Q \backslash P$(i.e. the data points in C are spread in this subspace).

For convenience of presentation, we will use a pair $(C, P)(P \neq \Phi)$ to denote a subspace cluster, where P is the non-empty set of dimensions associated with C. In particular, if $P = Q$, then this cluster is formed in the whole space of the data set.

Therefore if we have a cluster C with the associated set of dimensions P, then C is also a cluster in every subspace of Span(P). Hence, a good subspace clustering algorithm should be able to find clusters and the maximum associated set of dimensions. Consider, for example, a data set with 5 data points of 6 dimensional(given in Table 2). In this data set, it is obvious that $C = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$ is a cluster and the maximum set of dimensions should be $P = \{1, 2, 3, 4\}$. A good subspace clustering algorithm should be able to find this cluster and the maximum set of associated dimensions P.

Records	Values
$\mathbf{x}_1$	(A,A,A,A,B,B)
$\mathbf{x}_2$	(A,A,A,A,C,D)
$\mathbf{x}_3$	(A,A,A,A,D,C)
$\mathbf{x}_4$	(B,B,C,C,D,C)
$\mathbf{x}_5$	(B,B,D,D,C,D)

Table 2: A sample data set illustrates clusters embedded in subspaces of a high dimensional space.

We will introduce an objective function for subspace clustering, and then treat the clustering process as an optimization problem with the goal to minimize the objective functions.

3.1 Objective Function

The objective function for clustering and the objective function for determining the subspace of each cluster are defined in terms of compactness and separation.

Let C be a cluster with associated set of dimensions P. We define the compactness of C in Span(P) and the separation of C in Span(R)(where $R = Q \backslash P$) as follows:

$$\mathrm{Cp}(C, P) = \frac{\sum_{\mathbf{x}, \mathbf{y} \in C} \|\mathbf{x} - \mathbf{y}\|_P^2}{|P||C|^2}, \quad (1)$$

$$\mathrm{Sp}(C, R) = \begin{cases} \frac{\sum_{\mathbf{x}, \mathbf{y} \in C} \|\mathbf{x} - \mathbf{y}\|_R^2}{|R||C|^2}, & \text{if } R \neq \Phi; \\ 1, & \text{if } R = \Phi, \end{cases} \quad (2)$$

where $|P|$ and $|R|$ denote the number of elements in the sets P and R, respectively, and $\|\mathbf{x} - \mathbf{y}\|_P^2 = \sum_{j \in P} \delta(x_j, y_j)^2$ with $\delta(x, y) = 0$ if $x = y$, 1 otherwise, x_j and y_j are the jth dimension values of $\mathbf{x}$ and $\mathbf{y}$, respectively, $\|\mathbf{x} - \mathbf{y}\|_R^2$ is defined similarly. We shall also drop the index if the whole space is involved. Then from the definition, we have

$$\mathrm{Cp}(C, P) \in [0, 1] \text{ and } \mathrm{Sp}(C, R) \in [0, 1].$$

A natural criterion for the effectiveness of a subspace clustering is to simply sum up the compactness of each cluster

and then to minus the sum of separation of each cluster. This leads to the following objective function

$$F_{obj} = \sum_{j=1}^{k} (\mathrm{Cp}(C_j, P_j) + 1 - \mathrm{Sp}(C_j, R_j)). \quad (3)$$

Therefore, given the number of clusters k, our goal is to partition the data set into k non-overlapping groups such that the objective function F_{obj} defined in Equation (3) is minimized. Our algorithm is to find an approximation of the optimal partition.

In practice, we can simplify the formulas in Equation (1), Equation (2), and therefore simplify the objective function. To do this, we need to define the symbol table of a data set and the frequency table for each cluster according to the symbol table. Let A_j be the categorical variable of the jth dimension($1 \leq j \leq d$). We define its domain by $DOM(A_j) = \{A_{j1}, A_{j2}, ..., A_{jn_j}\}$ and we call $A_{jr}(1 \leq r \leq n_j)$ a state of the categorical variable A_j. Then a symbol table T_s of the data set is defined as follows:

$$T_s = (\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_d),$$

where $\mathbf{s}_j$ is a vector defined as $\mathbf{s}_j = (A_{j1}, A_{j2}, ..., A_{jn_j})^T$. Since there are possibly multiple states(or values) for a variable, a symbol table of a data set is usually not unique. For example, for the data set in Table 2, Table 3 is one of its symbol tables.

$$\begin{pmatrix} A & A & A & A & B & B \\ B & B & C & C & C & C \\ & & D & D & D & D \end{pmatrix}$$

Table 3: One of the symbol tables of the data set in Table 2.

The frequency table is computed according to a symbol table and it has exactly the same dimension as the symbol table. Let C be a cluster, then the frequency table $T_f(C)$ of cluster C is defined as

$$T_f(C) = (\mathbf{f}_1(C), \mathbf{f}_2(C), ..., \mathbf{f}_d(C)), \quad (4)$$

where $\mathbf{f}_j(C)$ is a vector defined as

$$\mathbf{f}_j(C) = (f_{j1}(C), f_{j2}(C), ..., f_{jn_j}(C))^T, \quad (5)$$

where $f_{jr}(C)(1 \leq j \leq d, 1 \leq r \leq n_j)$ is the number of data points in cluster C which take value A_{jr} at the jth dimension, i.e.

$$f_{jr}(C) = |\{\mathbf{x} \in C : x_j = A_{jr}\}|, \quad (6)$$

where x_j denotes the jth dimension of $\mathbf{x}$.

For a given symbol table of the data set, the frequency table of each cluster is unique according to that symbol table. For example, for the data set in Table 2, let (C, P) be a subspace cluster, where $C = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$ and $P = \{1, 2, 3, 4\}$, if we use the symbol table presented in Table 3, then the corresponding frequency table for the subspace cluster (C, P) is given in Table 4.

From the definition of frequency f_{jr} in Equation (6), we have the following equalities:

$$\sum_{r=1}^{n_j} f_{jr}(C) = |C|, \quad j = 1, 2, ..., d, \quad (7)$$

$$\begin{pmatrix} 3 & 3 & 3 & 3 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ & & 0 & 0 & 1 & 1 \end{pmatrix}$$

Table 4: The frequency table computed from the symbol table in Table 3.

for any subspace cluster (C, P).

We can now use the frequency table to simplify the formulas of compactness and separation. Let C be a cluster with the set P of dimensions associated, $T_f(C)$ be its frequency table. Since the square of the simple matching distance is equal to itself and using Equation (7), after a simple manipulation, we have

$$\sum_{\mathbf{x},\mathbf{y} \in C} \|\mathbf{x} - \mathbf{y}\|_P^2 = |P| \cdot |C|^2 - \sum_{j \in P} \|\mathbf{f}_j(C)\|^2, \quad (8)$$

where $\mathbf{f}_j(C)$ is defined in Equation (5) and $\|\cdot\|$ is the usual Euclidean norm. Note that the same notation is used for the Euclidean norm of a point in an Euclidean space and for the matching distance (defined below equation (2)) of two points in the original data space, this should be easily distinguished from the context.

Thus from Equation (8), we obtain the following simplified formulas of compactness and separation:

$$\mathrm{Cp}(C, P) = 1 - \frac{\sum_{j \in P} \|\mathbf{f}_j(C)\|^2}{|P||C|^2}, \quad (9)$$

$$\mathrm{Sp}(C, R) = \begin{cases} 1 - \frac{\sum_{j \in R} \|\mathbf{f}_j(C)\|^2}{|R||C|^2}, & \text{if } R \neq \Phi; \\ 1, & \text{if } R = \Phi. \end{cases} \quad (10)$$

3.2 Algorithm

We have defined the objective function (3), and introduced a simple way to calculate this function. Our goal is to partition the data set into k non-overlapping groups such that the function in Equation (3) is minimized. We now introduce our algorithm.

In the first step of this algorithm, we need to initialize the partition, i.e. given the number of clusters k, we need to partition the data set into k non-overlapping groups. There are many ways to do this. One way is to partition the data set into k non-overlapping groups randomly, but this is not efficient for the clustering task in the next steps. Another way, that we shall take, is to compute the proximity matrix of the data set, then choose k most dissimilar data points(See Section 3.3) as seeds according to the proximity matrix, and then assign the remaining data points to the nearest seed. Since computing the proximity matrix for large data set is impractical, we can first draw a sample(usually of small size) from the data set, and then compute the proximity matrix for the sample. We will discuss the initialization phase, in Section 3.3 and Section 3.4, in detail.

After the initialization phase, we begin to optimize the partition such that the objective function (3) is minimized. To optimize the partition, the algorithm will move a point from its current cluster to another cluster if the movement can decrease the objective function. In practice, the algorithm will stop if there is no further change of cluster memberships. We will discuss the optimization phase in Section 3.5 in detail.

Algorithm 3.1 The pseudo code of SUBCAD.

Require: D - Data Set, k - Number of Clusters;
Ensure: $2 \leq k \leq |D|$;
1: **if** D is a large data set **then**
2: Draw a sample from D;
3: **end if**
4: Compute the proximity matrix from the whole data set or the sample;
5: Pick k most dissimilar data points as seeds;
6: Assign the remaining data points to the nearest seed;
7: **repeat**
8: **for** $i = 1$ to $|D|$ **do**
9: Let (C_l, P_l) be the subspace cluster that contains $\mathbf{x}_i$;
10: **for** $m = 1, m \neq l$ to k **do**
11: **if** Inequality (18) is true **then**
12: Move $\mathbf{x}$ from C_l to C_m;
13: Update subspaces P_l and P_m;
14: **end if**
15: **end for**
16: **end for**
17: **until** No further change of the cluster memberships;
18: Output results.

In summary, the algorithm consists of two phases: the initialization phase and the optimization phase. In the following sections, we will discuss the criteria of moving a data point from its current cluster to another cluster and the criteria of determining the subspace associated with each cluster.

3.3 The Initialization Phase

In the initialization phase, we initialize the partition for the optimization. In partitioning clustering algorithms, the initial partition is very important. Good initial partition leads to fast convergence of the algorithm. Some initialization methods have been proposed in the literature of clustering, such as cluster-based method [7], *kd*-tree based method [21]. Denote by k the number of clusters in this algorithm, we first pick k most dissimilar data points as seeds, then we assign the remaining data points to the nearest seed. To make the algorithm clear, we introduce:

DEFINITION 1. *Let D be a data set, k be a positive integer such that $k \leq |D|$. We say that $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_k \in D$ are k most dissimilar data points of the data set D if the following condition is satisfied*

$$\min_{1 \leq r < s \leq k} \|\mathbf{x}_r - \mathbf{x}_s\| = \max_{E \in \mathcal{F}} \min_{\mathbf{x}, \mathbf{y} \in E} \|\mathbf{x} - \mathbf{y}\|, \tag{11}$$

where $\mathcal{F}$ is the class that contains all subsets E of D such that $|E| = k$, i.e.

$$\mathcal{F} = \{E : E \subseteq D, |E| = k\},$$

and $\|\mathbf{x} - \mathbf{y}\|$ is the distance between $\mathbf{x}$ and $\mathbf{y}$. For convenience, we use $X(k, D) (k \leq |D|)$ to denote the set of k most dissimilar data points of the data set D. □

Note that the set of k most dissimilar data points of a data set is not unique if $k < |D|$, i.e. $X(k, D)$ $(k < |D|)$ is not unique. For example, in the data set listed in Table 2, let $D = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\}$ and $k = 2$, then according to Definition 1, $X(2, D)$ can be $\{\mathbf{x}_1, \mathbf{x}_4\}$ or $\{\mathbf{x}_2, \mathbf{x}_4\}$. Of course, if $k = |D|$, then $X(|D|, D) = D$ is unique.

Since there are total $\begin{pmatrix} n \\ k \end{pmatrix}$ elements in $\mathcal{F}$, when n is large, it is impractical to enumerate all sets in $\mathcal{F}$ to find a set of k most dissimilar data points. Thus we use an approximation algorithm to find a set of k data points that is near the set of k most dissimilar data points. The basic idea is to choose k initial data points, then continuously replace the bad data points with good ones until no further changes.

More specifically, let $D = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ be a data set or a sample from a data set. First, we let $X(k, D)$ be $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_k\}$, let $\mathbf{x}_r, \mathbf{x}_s (1 \leq r < s \leq k)$ be such that

$$\|\mathbf{x}_r - \mathbf{x}_s\| = \min_{\mathbf{x}, \mathbf{y} \in X(k,D)} \|\mathbf{x} - \mathbf{y}\|. \tag{12}$$

Secondly, for each of the data point $\mathbf{x} \in D \backslash X(k, D)$, let

$$S_r = \min_{\mathbf{y} \in X(k,D) \backslash \{\mathbf{x}_s\}} \|\mathbf{x} - \mathbf{y}\|, \tag{13}$$

$$S_s = \min_{\mathbf{y} \in (X(k,D) \backslash \{\mathbf{x}_r\})} \|\mathbf{x} - \mathbf{y}\|. \tag{14}$$

Then if $S_r > \|\mathbf{x}_r - \mathbf{x}_s\|$, we let $X(k, D) \cup \{\mathbf{x}\} \backslash \{\mathbf{x}_s\}$ replace $X(k, D)$; if $S_s > \|\mathbf{x}_r - \mathbf{x}_s\|$, we let $X(k, D) \cup \{\mathbf{x}\} \backslash \{\mathbf{x}_r\}$ replace $X(k, D)$.

3.4 Sampling for large data sets

In the initialization phase, we need to select k seeds from the data set. If the data set is very large, then to compute the proximity matrix of the whole data set is impractical. To make the algorithm scale to large data set, we will draw samples from the original data set and choose seeds from the samples. In this section, we will discuss sampling methods for large data sets in the initialization phase.

Many algorithms for drawing a sample randomly from data sets have been designed, such as density biased sampling [20], random sampling with a reservoir [16; 23]. Also as for how to choose the sample size, we can use Chernoff bounds [18; 12] to determine the sample size such that the sample contains as least a certain amount data points from an arbitrary cluster with a high probability.

Let $C_b(k, n)$ be the minimum size of sample S such that every cluster has more than ξ data points in the sample with probability $1 - \delta$, then $C_b(k, n)$ can be computed from the following equation [25; 12]:

$$C_b(k, n) = \xi k \rho + k \rho \log\left(\frac{1}{\delta}\right) + k\rho \sqrt{\left(2\xi + \log\frac{1}{\delta}\right) \log\frac{1}{\delta}}, \tag{15}$$

where ρ is given by

$$\rho = \frac{n}{k \cdot |C_{min}|},$$

and C_{min} is the smallest cluster in the partition.

Note that for a small data set, sampling is not necessary. When sampling is necessary depends on the machine on which the algorithm runs. But in general, we can determine whether or not to draw a sample as follows. If $n > C_b(k, n)$, then we take sample of size $C_b(k, n)$; if $n < C_b(k, n)$, we just use the whole data set to compute proximity matrix. If $n > C_b(k, n)$, from Equation (15) and notice that $\rho \geq 1$, we

have

$$n > k\left(\xi + \log\left(\frac{1}{\delta}\right) + \sqrt{\left(2\xi + \log\frac{1}{\delta}\right)\log\frac{1}{\delta}}\right). \quad (16)$$

Thus, if inequality (16) is true, we draw samples of size $C_b(k,n)$ from the original data sets. We take $\xi = 50$ and $\rho = 0.01$ for default values

3.5 Optimization Phase

In the optimization phase of the algorithm, we need to reassign the data points in order to minimize the objective function (3) and update the subspaces associated with these clusters whose memberships are changed.

Let $(C_1, P_1), (C_2, P_2), ..., (C_k, P_k)$ be a partition of the data set D, let $\mathbf{x}$ be a data point in the subspace cluster (C_l, P_l). To achieve the membership changing rules, we use "exact assignment test" [24] technique in our algorithm. We will move $\mathbf{x}$ from subspace cluster (C_l, P_l) to another subspace cluster (C_m, P_m) $(m \neq l)$ if the resulted cost function decreases, i.e. if the following inequality is true:

$$\begin{aligned}\sum_{i=1}^{k}(\mathrm{Cp}(C_i) + 1 - \mathrm{Sp}(C_i)) &> \sum_{i=1, i\neq l, m}^{k}\mathrm{Cp}(C_i)\\ + \quad &\mathrm{Cp}(C_l - \mathbf{x}) + 1 - \mathrm{Sp}(C_l - \mathbf{x}) + \mathrm{Cp}(C_m + \mathbf{x})\\ + \quad &1 - \mathrm{Sp}(C_m + \mathbf{x}), \quad (17)\end{aligned}$$

where $C_l - \mathbf{x}$ means $C_l \backslash \{\mathbf{x}\}$, $C_m + \mathbf{x}$ means $C_m \cup \{\mathbf{x}\}$. Inequality (17) is equivalent to

$$\begin{aligned}&\mathrm{Cp}(C_l) - \mathrm{Cp}(C_l - \mathbf{x}) - \mathrm{Sp}(C_l) + \mathrm{Sp}(C_l - \mathbf{x}) >\\ &-\mathrm{Cp}(C_m) + \mathrm{Cp}(C_m + \mathbf{x}) + \mathrm{Sp}(C_m) - \mathrm{Sp}(C_m + \mathbf{x}). \quad (18)\end{aligned}$$

Hence if Inequality (18) is true, the data point $\mathbf{x}$ will be moved from C_l to C_m. After this movement, the sets of subspace dimensions P_l and P_m will of course be updated(See Section 4).

Let the symbol table of the data set D be T_s, and let $r_j (j = 1, 2, ..., d)$ be the subscript such that $\mathbf{x}_j = A_{jr_j}$. Then the frequency table of $C_l - \mathbf{x}$ is the same as the frequency table of C_l except for the terms $f_{jr_j}(C_l - \mathbf{x}) = f_{jr_j}(C_l) - 1 (j = 1, 2, ..., d)$; the frequency table of $C_m + \mathbf{x}$ is the same as the frequency table of C_m except for the terms $f_{jr_j}(C_m + \mathbf{x}) = f_{jr_j}(C_m) + 1$. These relationships enable us to rewrite Inequality (18) in a more compact form, omitted here due to limitation of spaces.

4. DETERMINE SUBSPACES

Let (C, E) be a subspace cluster of a d-dimensional data set D. In order to determine the set P of subspace dimensions associated with C, we define an objective function whose domain is all the subsets of $Q = \{1, 2, ..., d\}$ as follows:

$$F(C, E) = \mathrm{Cp}(C, E) + 1 - \mathrm{Sp}(C, Q\backslash E), \quad \Phi \neq E \subseteq Q, \quad (19)$$

where $\mathrm{Cp}(C, E)$ and $\mathrm{Sp}(C, Q\backslash E)$ are the compactness and separation of cluster C under the subspace dimensions set E.

Our general idea to determine the set P associated with cluster C is to find a P such that the objective function defined in Equation (19) is minimized. Also from Equation (8), if $P \neq \Phi$ or $P \neq Q$, we can write the objective function in Equation (19) as

$$F(C, E) = 1 - \frac{\sum_{j\in E}\|\mathbf{f}_j(C)\|^2}{|E||C|^2} + \frac{\sum_{j\in Q\backslash E}\|\mathbf{f}_j(C)\|^2}{|Q\backslash E||C|^2}, \quad E \subseteq Q, \quad (20)$$

where $R = Q\backslash P$.

We now establish some useful properties of the objective function.

THEOREM 1 (CONDITION OF CONSTANCE). *The objective function defined in Equation (19) is constant for any subset E of Q if and only if*

$$\|\mathbf{f}_1(C)\| = \|\mathbf{f}_2(C)\| = \cdots = \|\mathbf{f}_d(C)\|. \quad (21)$$

In addition, if the objective function is a constant, then it is equal to 1.

¿From the definition of the objective function $F(C, E)$, the proof of the above theorem is straightforward and is thus omitted. By Theorem 1, if the objective function is constant, then the objective function is minimized at any subset of Q. In this case, we define the subspace dimensions associated with C to be Q. If the objective function is not constant, then we define the subspace dimensions associated with C to be the set $P \subseteq Q$ that minimizes the objective function. In fact, we can prove late that such a set P is unique if the objective is not constant. Hence, we have the following definition of subspace associated with each cluster.

DEFINITION 2. *Let C be a cluster, then the set P of subspace dimensions associated with C is define as follows:*

1. *If the objective function $F(C, E)$ is constant for any $E \subseteq Q$, then let $P = Q$;*
2. *If the objective function $F(C, E)$ is not constant, then P is defined as*

$$P = \arg\max_{E\in\mathcal{E}}|E|, \quad (22)$$

where $\mathcal{E}$ is defined as

$$\mathcal{E} = \{O : F(C, O) = \min_{E\in\aleph} F(C, E), O \in \aleph\}, \quad (23)$$

and $\aleph$ is defined as

$$\aleph = \{E : E \subset Q, E \neq \Phi, E \neq Q\}. \quad (24)$$

REMARK 1. *¿From Definition 2, the set P defined in Equation (22) is non-empty. Moreover, if the objective function $F(C, E)$ is not constant, then the set P is a true subset of Q, i.e. $P \subsetneqq Q$.*

Below we will prove that if the objective function $F(C, E)$ defined in Equation (20) is not constant, then the set P defined in Equation (22) is unique. To do this, we first derive some properties of the set P in Equation (22).

THEOREM 2. *Let (C, P) be a subspace cluster of data set D in a d-dimensional space($d > 2$), let P be defined in Equation (22). Then*

1. *for a given $r \in P$, if there exists a $s (1 \leq s \leq d)$ such that $\|\mathbf{f}_s(C)\| > \|\mathbf{f}_r(C)\|$, then $s \in P$;*
2. *for a given $r \in R$, if there exists a $s (1 \leq s \leq d)$ such that $\|\mathbf{f}_s(C)\| < \|\mathbf{f}_r(C)\|$, then $s \in R$, where $R = Q\backslash P$.*

Theorem 2 can be proved by way of contradiction. To prove the first part, for example, suppose $s \notin P$ and let $P_{new} = P \cup \{s\}\backslash\{r\}$, $R_{new} = R \cup \{r\}\backslash\{s\}$, then one can show that $F(C, P_{new}) < F(C, P)$, a contradiction. A detailed proof can be found in [10]
Let (C, P) be a subspace cluster, where P is defined in Equation (22). Then from Theorem 2, there exists no $r, s(1 \leq r, s \leq d)$ such that $r \in P, s \in R$ and $\|\mathbf{f}_r(C)\| < \|\mathbf{f}_s(C)\|$. Hence we have the following corollary:

COROLLARY 3 (MONOTONICITY). *Let (C, P) be a subspace cluster of D, where P is the set of subspace dimensions defined in Equation (22) and let $T_f(C)$ be the frequency table of C, then for any $r \in P$ and $s \in R(R = Q\backslash P)$, we have*

$$\|\mathbf{f}_r(C)\| \geq \|\mathbf{f}_s(C)\|. \tag{25}$$

Now we consider the case where there exist $r \in P$ and $s \in R$ such that $\|\mathbf{f}_r(C)\| = \|\mathbf{f}_s(C)\|$.

THEOREM 4. *Let (C, P) be a subspace cluster of d dimensional data set $D(d > 2)$, where P is defined in Equation (22), let $T_f(C)$ be the frequency table defined in Equation (4), let $r, s(1 \leq r, s \leq d)$be given so that $\|\mathbf{f}_s(C)\| = \|\mathbf{f}_r(C)\|$. Then either $r, s \in P$ or $r, s \in R$, i.e. r, s must be in the same set of P or R, where $R = Q\backslash P$.*

Theorem 4 can be proved using a similar argument for Theorem 2, details can be found in [10]. ¿From Corollary 3 and Theorem 4, we have the following:

COROLLARY 5. *Let (C, P) be a subspace cluster of D, where P is the set of subspace dimensions defined in Equation (22), and let $T_f(C)$ be the frequency table of C. If the objective function $F(C, E)$ is not constant, then for any $r \in P$ and $s \in R(R = Q\backslash P)$, we have*

$$\|\mathbf{f}_r(C)\| > \|\mathbf{f}_s(C)\|. \tag{26}$$

Now using Corollary 5, we can prove the uniqueness of the set P defined in Equation (22).

THEOREM 6 (UNIQUENESS OF SUBSPACE). *Let $F(C, E)$ be the objective function defined in Equation (19) and let P be the set defined in Equation (22). If the objective function $F(C, E)$ is not constant, then the set P is unique.*

Also from the Corollary 5 and the Theorem 6, we have the following theorem, based on which we can design a very fast algorithm to find the set of subspace dimensions associated with each cluster. The detailed proof of the following theorem can also be found in [10].

THEOREM 7 (CONTIGUITY). *Let (C, P) be a subspace cluster of D, where P is the set of subspace dimensions defined in Equation (22). Let $T_f(C)$ be the frequency table of C, and let $i_1, i_2, ..., i_d$ be a combination of $1, 2, ..., d$ such that*

$$\|\mathbf{f}_{i_1}(C)\| \geq \|\mathbf{f}_{i_2}(C)\| \geq \cdots \geq \|\mathbf{f}_{i_d}(C)\|.$$

Finally, let G_s be the set of subscripts defined as

$$G_s = \left\{t : \|\mathbf{f}_{i_t}(C)\| \neq \left\|\mathbf{f}_{i_{t+1}}(C)\right\|, 1 \leq t \leq d-1\right\}. \tag{27}$$

If the objective function defined in Equation (19) is not constant, then the set of subspace dimensions P defined in Equation (22) must be one of the P_k's$(k = 1, 2, ..., |G_s|)$ defined as follows:

$$P_k = \{i_t : t = 1, 2, ..., g_k\}, \quad k = 1, 2, ..., |G_s|, \tag{28}$$

where $g_1 < g_2 < \cdots < g_{|G_s|}$ are elements of G_s.

Based on Theorem 7, we can find the set of subspace dimensions P for a cluster C very fast. There are totally $2^d - 1$ non-empty subsets of Q, it is impractical to find an optimal P by enumerating these $2^d - 1$ subsets. Based on Theorem 7, we can design a fast algorithm for determining the set of subspace dimensions.

5. EXPERIMENTS

To evaluate the performance of the algorithm, we implemented our algorithm on Sun Blade 1000 workstation using GUN C++ compiler. In this section, we shall use experimental results to show the clustering performance of SUBCAD.

5.1 Data sets

We chose not to run our experiments on synthetic datasets, not only because synthetic data sets may not well represent real world data [13], but also because there is no well established categorical data generation method. Therefore, instead of generating synthetic data to validate the clustering algorithm we choose three real world data sets obtained from UCI Machine Learning Repository [6]. All these data sets have class labels assigned to the instances.

5.1.1 Soybean data

The soybean data set has 47 records each of which is described by 35 attributes. Each record is labelled as one of the 4 diseases: diaporthe stem rot, charcoal rot, rhizoctonia root rot and phytophthora rot. Except for the phytophthora rot which has 17 instances, all other diseases have 10 instances each. Since there are 14 attributes that have only one category, we only selected other 21 attributes for the purpose of clustering.

5.1.2 Wisconsin breast cancer data

The Wisconsin breast cancer data set has total 699 records, each of which is described by 10 categorical values. There are 16 records that have missing values. Since our algorithms do not deal with missing values, we delete the 16 records from the data set and use the remaining 683 records for testing.

5.1.3 Congressional voting data

The Congressional voting data set includes votes for each of the U.S. House of Representatives Congressmen on the 16 key votes identified by the CQA. It has 435 instances(267 democrats, 168 republicans) and some of the instances have missing values, we denote the missing value by "?" and treat it as the same as other values.

	Cluster 1	Cluster 2	Cluster 3	Cluster 4
D	8	0	0	2
C	0	0	10	0
R	0	0	0	10
P	0	16	0	1

Table 5: The misclassification matrix of the result obtained by applying SUBCAD to the soybean data, where D,C,R,P denote four different diseases.

5.2 Clustering results

To measure the qualities of clustering results of our clustering algorithms, we use clustering accuracy measure r defined as follows [13]:

$$r = \frac{\sum_{i=1}^{k} a_i}{n}, \tag{29}$$

where a_i is the number of instances occurring in both cluster i and its corresponding class, and n is the number of instances in the data set.

Table 5 shows the misclassification matrix of the clustering result by applying our algorithm to the soybean data. According to the clustering accuracy measure r defined in Equation (29), the clustering accuracy is

$$r_a = \frac{8 + 10 + 10 + 16}{47} = 0.9362. \tag{30}$$

Clusters	Sets of Subspace Dimensions
Cluster 1	$Q\backslash\{1\}$
Cluster 2	$Q\backslash\{1, 6\}$
Cluster 3	$Q\backslash\{1, 6, 10\}$
Cluster 4	$Q\backslash\{1, 6\}$

Table 6: The set of subspace dimensions associated with each cluster for the soybean data, where $Q = \{1, 2, ..., 21\}$.

Table 6 gives the subspace dimensions associated with each cluster in Table 5. Cluster 1 is formed in a 20-dimensional space, Cluster 2 and Cluster 4 are formed in the same subspace which is 19-dimensional, Cluster 3 is formed in a 18-dimensional subspace. The result shows that the clusters are formed almost in the whole space. Hence if apply conventional clustering algorithms(i.e. not subspace clustering algorithms), such as k-Modes[13], to the soybean data, one will get almost the same results.

Table 7 shows the misclassification matrix of the clustering result by applying our algorithm to the Wisconsin breast cancer data. Similarly, the clustering accuracy is

$$r_b = \frac{440 + 158}{683} = 0.8755. \tag{31}$$

	Cluster 1	Cluster 2
Benign	4	440
Malignant	158	81

Table 7: The misclassification matrix of the result obtained by applying SUBCAD to the Wisconsin breast cancer data.

Clusters	Sets of Subspace Dimensions
Cluster 1	$\{7\}$
Cluster 2	$Q\backslash\{1\}$

Table 8: Sets of subspace dimensions associated with each cluster for the Wisconsin breast cancer data, where $Q = \{1, 2, ..., 10\}$.

Table 8 gives the subspace dimensions associated with each cluster of the Wisconsin breast data. One cluster is formed in a 1-dimensional subspace, while the other one is formed in a 9-dimensional subspace. Under our objective function for determining subspaces, one cluster tends to have low dimensionality while the other tends to have high dimensionality.

Table 9 shows the misclassification matrix of the clustering result by applying our algorithm to the congressional voting data. Similarly, the clustering accuracy is

$$r_c = \frac{253 + 147}{435} = 0.9195. \tag{32}$$

	Cluster 1	Cluster 2
democrat	14	253
republican	147	21

Table 9: The misclassification matrix of the result obtained by applying SUBCAD to the congressional voting data.

Clusters	Sets of Subspace Dimensions
Cluster 1	$Q\backslash\{2, 16\}$
Cluster 2	$\{3, 4\}$

Table 10: Sets of subspace dimensions associated with each cluster for the congressional voting data, where $Q = \{1, 2, ..., 16\}$.

The subspace dimensions associated with the clusters of congressional voting data is given in Table 10. Similar to the clustering results of Wisconsin breast cancer data, one cluster of the congressional voting data has a low dimensionality while another has a high dimensionality.

6. CONCLUSION

In this paper we presented SUBCAD, an algorithm for subspace clustering high dimensional categorical data. We treat both the process of clustering and the process of determining the subspaces of clusters as a process of optimizing a certain cost function. The idea of optimization in determining the subspace of each cluster enables us to rapidly identify the subspaces in which the clusters are embedded. We tested the algorithm using various real world data sets from UCI Machine Learning Repository [6], with very good clustering accuracy. It should be mentioned that for some data sets, the algorithm tends to find some clusters in high-dimensional subspaces in conjunction with other clusters in low-dimensional subspaces. The rate of convergence depends on the size of the date set, as shown in our simulation on the Connect-4 Database (67557 records, each of which is described by 42 attributes). Furthermore, SUBCAD requires the number of clusters as an input parameter, and hence how to incorporate the existing methods of selecting this parameter into SUBCAD remains an interesting and challenging problem.

Acknowledgements

This research was partially supported by NSERC(Natural Sciences and Engineering Research Council of Canada), by CRC(Canada Research Chairs) Program, and by Generation 5.

7. REFERENCES

[1] C. Aggarwal, J. Wolf, P. Yu, C. Procopiuc, and J. Park. Fast algorithms for projected clustering. In *Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, pages 61–72. ACM Press, 1999.

[2] C. Aggarwal and P. Yu. Finding generalized projected clusters in high dimensional spaces. In W. Chen, J. F. Naughton, and P. A. Bernstein, editors, *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA*, volume 29, pages 70–81. ACM, 2000.

[3] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *SIGMOD Record ACM Special Interest Group on Management of Data*, pages 94–105, 1998.

[4] M. Anderberg. *Cluster analysis for applications*. Academic Press, New York, 1973.

[5] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "nearest neighbor" meaningful? In C. Beeri and P. Buneman, editors, *Database Theory - ICDT '99, 7th International Conference, Jerusalem,Israel, January 10-12, 1999, Proceedings*, volume 1540 of *Lecture Notes in Computer Science*, pages 217–235. Springer, 1999.

[6] C. Blake and C. Merz. UCI repository of machine learning databases, 1998. `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

[7] P. Bradley and U. Fayyad. Refining initial points for K-Means clustering. In *Proc. 15th International Conf. on Machine Learning*, pages 91–99. Morgan Kaufmann, San Francisco, CA, 1998.

[8] Y. Cao and J. Wu. Projective ART for clustering data sets in high dimensional spaces. *Neural Networks*, 15(1):105–120, January 2002.

[9] C. Cheng, A. Fu, and Y. Zhang. Entropy-based subspace clustering for mining numerical data. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 84–93. ACM Press, 1999.

[10] G. Gan. Subspace clustering for high dimendional categorical data. Master's thesis, Department of Mathematics and Statistics, York University, Toronto, Canada, October 2003.

[11] S. Goil, H. Nagesh, and A. Choudhary. MAFIA: Efficient and scalable subspace clustering for very large data sets. Technical Report CPDC-TR-9906-010, Center for Parallel and Distributed Computing, Department of Electrical & Computer Engineering, Northwestern University, June 1999.

[12] S. Guha, R. Rastogi, and K. Shim. CURE: an efficient clustering algorithm for large databases. In *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pages 73–84. ACM Press, 1998.

[13] Z. Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 2:283–304, 1998.

[14] A. Jain and R. Dubes. *Algorithms for Clustering Data*. Prentice Hall,Englewood Cliffs, New Jersey, 1988.

[15] L. Kaufman and P. Rousseeuw. *Finding Groups in Data–An Introduction to Cluster Analysis*. Wiley series in probability and mathematical statistics. John Wiley & Sons, Inc., New York, 1990.

[16] K. Li. Reservoir-sampling algorithms of time complexity o(n(1 + log(n/n))). *ACM Transactions on Mathematical Software (TOMS)*, 20(4):481–493, 1994.

[17] B. Liu, Y. Xia, and P. Yu. Clustering through decision tree construction. In *Proceedings of the ninth international conference on Information and knowledge management*, pages 20–29, McLean, Virginia, USA, 2000. ACM Press.

[18] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, New York, 1995.

[19] H. Nagesh, S. Goil, and A. Choudhary. A scalable parallel subspace clustering algorithm for massive data sets. In *2000 International Conference on Parallel Processing (ICPP'00)*, pages 477–486, Washington - Brussels - Tokyo, August 2000. IEEE.

[20] C. Palmer and C. Faloutsos. Density biased sampling: an improved method for data mining and clustering. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 82–92. ACM Press, 2000.

[21] D. Pelleg and A. Moore. Accelerating exact k-means algorithms with geometric reasoning. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 277–281. ACM Press, 1999.

[22] C. Procopiuc, M. Jones, P. Agarwal, and T. Murali. A monte carlo algorithm for fast projective clustering. In *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, pages 418–427. ACM Press, 2002.

[23] J. Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1):37–57, 1985.

[24] D. Wishart. k-means clustering with outlier detection, mixed variables and missing values. In M. Schwaiger and O. Opitz, editors, *Exploratory Data Analysis in Empirical Research*, pages 216–226. Springer, 2002.

[25] K. Woo and J. Lee. *FINDIT: a fast and intelligent subspace clustering algorithm using dimension voting*. PhD thesis, Korea Advanced Institue of Science and Technology, Department of Electrical Engineering and Computer Science, 2002.

[26] J. Yang, W. Wang, H. Wang, and P. Yu. δ-clusters: capturing subspace correlation in a large data set. *Data Engineering, 2002. Proceedings. 18th International Conference on*, pages 517 –528, 26 Feb.-1 March 2002.

[27] K. Yeung and W. Ruzzo. Principal component analysis for clustering gene expression data. *Bioinformatics*, 17(9):763–774, September 2001.