Breaking Down Dimensionality: Effective and Efficient Feature Selection for High-Dimensional Clustering

Diansheng Guo, Mark Gahegan, Donna Peuquet, Alan MacEachren

GeoVISTA Center & Department of Geography, Pennsylvania State University
302 Walker Building, University Park, PA 16802, USA
1-814-865-3433, http://www.geovista.psu.edu
dguo@psu.edu, mark@geog.psu.edu, peuquet@geog.psu.edu, maceachren@psu.edu

ABSTRACT

The *high dimensionality*[1] of data sets can cause serious problems for all data analysis methods, especially for exploratory clustering techniques. First, irrelevant or noisy attributes often exist in a data set. Using all original features to derive clusters can be ineffective and even counter-productive (they may hide rather than uncover clusters). Second, clusters may exist in different *subspaces*[2], i.e., different sets of points may cluster better in different subsets of the original features. Therefore, an effective feature selection method for high-dimensional clustering should be able to find all subspaces (not just a single "optimal" subspace) that contain strong clusters. To effectively and efficiently identify interesting subspaces, we propose a novel feature selection method for clustering (or, in general, exploring) high-dimensional data. The method is based on a generic measure of cluster tendency between dimensions and does not need to identify the actual clusters. A new calculation of the conditional entropy is developed to reliably measure the existence and the significance level of clusters (defined as a contiguous dense region of arbitrary shape) in a 2-D space. Then, given a high-dimensional data set (S) of dimensionality d and size n, a matrix of pair-wise conditional entropy values of all 2-D subspaces in S is derived. From the visualization of the entropy matrix, the user can easily get a holistic understanding of various relationships among dimensions. Interesting subspaces of more than two dimensions can be automatically extracted or interactively identified from the matrix. Experiments with synthetic data sets show that the method is very robust for correct identification of subspaces of various dimensionalities that contain clusters of various sizes. Its computational complexity is $O(d^2 n \log n)$.

Keyword: Data Mining, Feature Selection, Subspace Clustering, Cluster Ordering, Visualization

1. INTRODUCTION

With the increasing power of data collection and the increasing complexity of problems we are tackling, datasets to be analyzed are often very large (e.g., >10,000 observations) and have a high dimensionality (e.g., >50 attributes). Such data sets are commonly compiled from multiple data sources, which might be collected for different purposes. By putting them together for analysis, we are hoping to find some *unknown* multivariate relationships or patterns[3]. Clustering analysis is one of the most important techniques for identifying such relationships (Jain and Dubes 1988; Jain, Murty et al. 1999). To scale well with large data volumes, many efficient clustering methods

[1] The *dimensionality* of a data set is the number of features it contains. In this paper, *feature*, *dimension*, *attribute*, or *variable* can be used interchangeably

[2] A *subspace* is a subset of the features in the original data space.

[3] *Patterns* and *relationships* are used interchangeably in this paper. Specifically they refer to clusters of various shapes.

have recently been developed (Schikuta 1996; Zhang, Ramakrishnan et al. 1996; Alsabti and Sanjay Ranka 1998; Guha, Rastogi et al. 1998; Bradley and Fayyad 1999).

High-dimensional data sets can also cause serious problems for all data analysis methods, especially for clustering techniques (Hinneburg and Keim 1999). Intuitively, two facts are obvious about data sets of very high dimensionality. First, the quality and relevance of the original attributes can vary dramatically. Irrelevant or noisy attributes often exist in the data set. Therefore, distance functions that use all dimensions of the data can be ineffective and even counter-productive (they may hide rather than uncover clusters). Secondly, meaningful relationships that involve all original (maybe 50+) attributes are unlikely. The average density of points anywhere in the original data space is likely to be very low due to the "*curse of dimensionality*" (Hinneburg and Keim 1999; Duda, Hart et al. 2001).

One solution to this problem is to select a subset of dimensions that are highly correlated and then find clusters in the corresponding *subspace*. Traditional feature selection methods are used in the area of supervised classification (Liu and Motoda 1998). Recently several unsupervised feature selection methods have been developed to select an "optimal" subset of features (Dy and Brodley 2000; Dy and Brodley 2000), or produce a pool of "good" subsets of dimensions (Kim, Street et al. 2000), for unsupervised clustering. Clusters can also exist in different *subspaces*. In other words, different sets of points may cluster better in different subspaces, the dimensionality of which may also vary. Therefore it may be ineffective to find a single "optimal" subset of dimensions for all clusters (Agarwal, Procopiuc et al. 1999). Several subspace clustering methods have been developed to detect clusters residing in different subspaces (Agrawal, Gehrke et al. 1998; Cheng, Fu et al. 1999; Aggarwal and Yu 2000; Procopiuc, Jones et al. 2002).

Nevertheless, the identification of interesting subspaces (or feature subsets) that contain clusters remains a challenging research problem. First, existing subspace clustering methods all rely on a specific clustering algorithm and may even rely on several subjective input parameters of that clustering algorithm, e.g., the number of clusters, the dimensionality of clusters, various thresholds, etc., which are hard to configure. The analyst needs to run the subspace clustering procedure many times with different input parameters to get a feel of which result might be more reasonable. Second, it is not easy to get an overall understanding of various relationships among dimensions and their change from one significance level to another. It is also hard to see the impact of removing one or more dimensions on various relationships without running the procedure again. Third, each of these existing methods has some limitations, which will be explained in next section.

To improve the effectiveness, efficiency, and the ease for human interaction and understanding, a novel feature selection approach is proposed. This approach can help select subspaces without the actual identification of clusters. Once a subspace is selected, any full-dimensional clustering method can be used to search for clusters in that subspace. Other than clustering, this method can also be used to (1) inform various high-dimensional visualization techniques to focus on a subspace for better views; and 2) add, remove, and/or extract attributes to compile a better data set for analysis (not limited to clustering).

The subspace selection approach is based on a matrix of pair-wise conditional entropy values of 2-D data spaces. A new calculation of conditional entropy is developed to reliably measure the existence and the significance level of clusters (defined as a contiguous dense area of arbitrary shape) in a 2-D space. From the entropy matrix, the user can easily get an understanding of the overall picture of various relationships among dimensions without giving any subjective parameter. Subspaces of more than two dimensions can be automatically extracted given an entropy threshold, which is easy to configure with the visual display of the matrix. The user can also interactively form a subspace based on both the entropy matrix and his/her expertise about the application problem. Experiments with synthetic data sets show that the approach is very robust for correct identification of subspaces of various dimensionalities that contain clusters of various sizes.

The rest of the paper is organized as follows. The next section will introduce related research directed to tackling the high dimensionality for clustering problems. Section 3 will present the calculation of a conditional entropy value for a 2-D data space. Then section 4 will introduce how to search interesting multidimensional subspaces based on the entropy matrix, automatic algorithms, and human interaction with the help of visualization. Section 5 includes experiments to demonstrate the ability of the subspace selection approach. The last section is conclusion and discussion.

2. RELATED WORK

The problem of high dimensionality is often tackled by requiring the user (who should be an expert on the application problem) to specify a subspace (or several subspaces) for cluster analysis. However, the user's identification of subspaces is error-prone. Moreover, depending on the user to choose subspaces will make it impossible to find *unexpected* patterns, while finding such *unexpected* patterns is one of the main purposes of data mining and knowledge discovery (Fayyad, Piatetsky-Shapiro et al. 1996). Neither the expert nor exhaustive enumerations (a brute-force solution) can solve the problem of high dimensionality alone.

2.1. Dimension Reduction Techniques

One way to address the problem of high dimensionality is to apply a dimension reduction method to the data set (Duda, Hart et al. 2001). Methods such as principle component analysis (PCA) transform the original data space into a low dimensional space by forming dimensions that are linear combinations of given attributes. While these techniques may succeed in reducing dimensionality and are useful for information compression and classification problems, they have several shortcomings with respect to clustering. First, the new dimensions can be difficult to interpret, making it even harder to understand clusters formed by new dimensions. Secondly, these techniques are not effective in identifying clusters that reside in different subspaces of the original data set. In other words, such dimension reduction approaches can only generate *one* single optimal subspace to represent the original data space. Third, PCA methods can only work well for linear relationships. Fourth, PCA uses all of the original features in the projection to a lower dimensional space. In other words, although dimensionality is reduced, the impact of every original dimension is more or less still there.

2.2. Feature Selection Methods

Feature selection methods are traditionally used to select a subset of dimensions for supervised classification problems (Liu and Motoda 1998). Recently several new feature selection methods have been developed to select an "optimal" subset of features for unsupervised clustering (Dy and Brodley 2000; Dy and Brodley 2000), or produce a pool of "good" dimension subsets for searching clusters (Kim, Street et al. 2000). Each of these methods centers around a specific clustering method, e.g. the expectation maximization (Dy and Brodley 2000) or the K-means (Kim, Street et al. 2000). However, it can be ineffective to rely on a specific clustering algorithm as a means to evaluate candidate subsets of dimensions. For example, K-means tends to discover equal-sized circular clusters and may fail to discover arbitrary-shaped patterns (e.g, in an extreme case, linear relationships), while an EM approach depends on a good initialization and also favors circular or ellipse clusters.

Scalability to high dimensionality (d) and very large data size (n) is another concern. Although efficient algorithms for K-means or EM-based clustering have been developed (Alsabti and Sanjay Ranka 1998; Bradley, Fayyad et al. 1998; Pelleg and Moore 1998), repeatedly using

such clustering algorithms to evaluate a large number of candidates (i.e., subsets of dimensions) can still cause computational efficiency problems, especially when both d and n are large.

2.3. Subspace Clustering Methods

Several subspace clustering methods can detect clusters residing in different subspaces (i.e., subsets of the original dimensions). No new dimension is generated, which is important because original dimensions bear real meaning to the user. Each resultant cluster is associated with a specific subspace. CLIQUE (Agrawal, Gehrke et al. 1998), ORCLUS (Aggarwal and Yu 2000), and DOC (Procopiuc, Jones et al. 2002) are three representative methods for subspace clustering.

CLIQUE (Clustering In Quest) adopts a density-based approach to clustering: a cluster is a region that has higher density of points than its surrounding area. To approximate the density of data points, the data space is partitioned into a finite set of cells. The *coverage* of a cell is the number of points it contains. CLIQUE partitions each dimension into the same number of equal length intervals. A cluster is defined as a maximal set of connected dense cells in a subspace. CLIQUE uses a bottom-up searching and pruning strategy—if a k-dimensional subspace does not have any dense cell, all subspaces that contain this subspace are pruned. The result of CLIQUE critically relies on two parameters: the number of intervals (ξ) and the density threshold (τ), which are hard to configure. Equal interval discretization suffers from outliers and extreme values and cannot adapt well to various data distributions.

ORCLUS (arbitrarily ORiented projected CLUSter generation) introduced the notion of generalized projected clusters. ORCLUS assumes two input parameters: the number of clusters (k) and the dimensionality (l) of subspaces (this means all subspaces that contain clusters are of the same dimensionality). These two input parameters actually are impossible to know beforehand. The overall algorithm for ORCLUS first picks k_0 initial points (called *seeds*) from the input data set. Then it runs a number of iterations, each of which does a sequence of merging operations to reduce the number of current clusters by the factor $\alpha < 1$ and reduce the dimensionality of current cluster C_i by factor $\beta < 1$. The algorithm terminates when the number of clusters has reduced to the specified number k and, *at the same time*, the dimensionality of each subspace associated with each cluster is also equal to the specified parameter l. To make sure that the reduction from k_0 to k occurs at the same time when the dimensionality is reduced from d to l, α and β need to satisfy $\log_{(1/\alpha)}(k_0/k) = \log_{(1/\beta)}(l_0/l)$. The performance of the algorithm degrades fast with increasing dimensionality. Other two drawbacks of ORCLUS are that: (1) the requirement that parameters of k and l are provided, and 2) the dependency between α and β.

DOC (Density-based Optimal projective Clustering) is a Monte Carlo algorithm that computes, with high probability, a good approximation of a projective cluster. The algorithm can be iterated and each iteration generates one new cluster. The iteration stops when some criterion is met. A projective cluster (C, D) with width w is an axis-parallel box, which has a maximum edge length of w and contains more than α portion of total points. DOC also needs a balance-factor β that represents the user's choice on the relative importance of the number of points versus the number of dimensions in a cluster. With at least ½ probability DOC can return an approximate solution if $1/(4d) \le \beta < ½$ and $0 < \alpha < 1$. The overall time complexity is $O(n \cdot d^{C+1})$, where $C = \log(2/\alpha) / \log(1/(2\beta))$. If $\alpha = 0.1$, $\beta = 0.25$, then the complexity is $O(n \cdot d^5)$. Parameters w, α, and β, are all hard to configure but all have dramatic influences on the result. The user needs to run the algorithm many times with different settings of one or more parameters to gain an overall understanding of the data set and the overall relationships among dimensions. Moreover, the accuracy of DOC is poor for clusters of low dimensionality (e.g., <10) (Procopiuc, Jones et al. 2002). However, for a real data analysis task, the ability to detect low-dimensional clusters within a high-dimensional data set is very important.

2.4. Cluster Tendency Measures

Clustering tendency (Jain and Dubes 1988; Jain, Murty et al. 1999) refers to the problem of deciding whether the data set exhibits a tendency to cluster into natural groups without the actual identification of those clusters. In other words, clustering tendency is a quantitative measurement that indicates whether a data set has clusters and how strong those clusters are. An examination of the clustering tendency is an important part of for the overall clustering procedure because many clustering algorithms will create clusters whether or not the data are naturally clustered or just random.

In this paper, a cluster is defined as a contiguous, dense region (in multidimensional space) of arbitrary shape. A partition of the data space is not required, i.e., the data may contain only one cluster. Thus a linear relationship can be regarded as a special case of a cluster, which has an elongated shape. From this point of view, three types of cluster tendency measure exist: covariance or correlation, Quadrat analysis and chi-square test, and entropy-based measures.

The *covariance* or *correlation* measure in statistics has been long used for testing the existence of a *linear* relationship between two dimensions. However, for a data space, there may exist strong clusters while the correlation value is very low since correlation only captures linear relationships. Spatial autocorrelation can be regarded as a measure of clustering in a 2-D (usually geographic) space (Zhang and Murayama 2000).

Quadrat analysis partitions a 2-D data space into rectangles of equal size, called quadrants, and counts the number of points falling in each quadrant (Jain and Dubes 1988). If the data set contains no significant cluster, the set of counts will follow a Poisson distribution under randomness. The Chi-square test (Snedecor and Cochran 1989), which has been widely used for testing the statistical significance of bivariate tabular association, can then be used here to test a hypothesis of randomness for those quadrant counts.

Conditional entropy is another measurement for detecting the mutual interaction between two dimensions (attributes) (Pyle 1999). Cheng and others (1999) also proposed an *entropy-based approach* for evaluating and pruning subspaces (Cheng, Fu et al. 1999).

3. IMPROVING CONDITIONAL ENTROPY TO MEASURE CLUSTER TENDENCY

To efficiently tackle the problem of high dimensionality, we start from 2-D subspaces of a high-dimensional dataset. A cluster is defined as a contiguous, dense region of an arbitrary shape. The definition for being "dense" varies in different research (Ester, Kriegel et al. 1996; Agrawal, Gehrke et al. 1998; Hinneburg and Keim 1999). Since here we need not to actually identify clusters, a clear-cut definition of "being dense" is not necessary. It is important to define "being denser". For two regions R_1 and R_2 of the same size, if R_1 has more points than R_2 does, then R_1 is *denser* than R_2. Given a high-dimensional data set and two different 2-D subspaces (S_1 and S_2) from it, if S_1 has smaller and denser regions that contain the majority of data points than S_2 does, then S_1 is more "*clustered*" than S_2.

A good cluster tendency measure should meet several requirements:

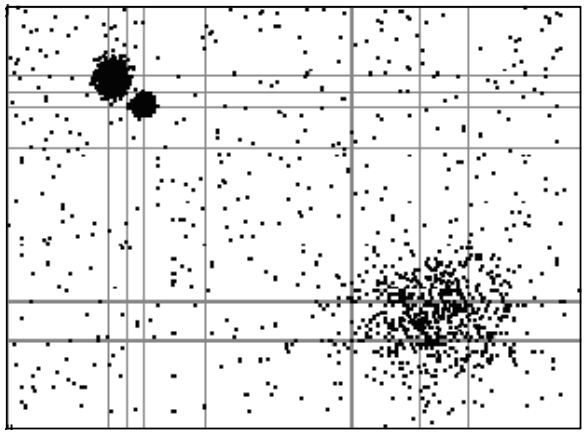
- It does not assume any particular cluster shape;
- It does not assume any particular cluster size (in terms of both the coverage and the range on each bound dimension);
- It does not assume any distribution of data points;
- It is robust with various dataset size;
- It is robust with outliers and extreme values;
- It is tolerant to a high portion of noise.

To meet all these requirements, the clustering tendency measure should be very generic. Both conditional entropy and Chi-square test (using the test value as a measure) potentially qualify.

However, they both need a discretization of the 2-D data space, which can critically influence the result. Traditionally they both use the equal-interval discretization method, which is not adaptive and robust to various data distributions. We propose a new calculation of the conditional entropy value and the Chi-square value using a *nested-means* discretization. In this paper we focus on conditional entropy. Chi-square values also work well but a comparison between a conditional-entropy-based approach and a chi-square-based approach needs more work and is beyond the scope of this paper.

3.1. Dimension Discretization

There are many existing discretization (classification) methods for single-dimensional data (Slocum 1999). CLIQUE, ENCLUS and Quadrat analysis all adopt the Equal-Interval (EI) method. However, EI cannot adapt well to various data distributions and often fail to capture patterns at different levels. We choose the Nested-Mean (NM) method (*figure 1*) to improve the effectiveness (Guo, Peuquet et al. 2002).



100	**294**	14	2	22	18	5	20
206	**547**	17	1	7	0	2	6
19	46	**336**	**268**	1	2	2	2
10	2	**328**	**273**	17	5	4	7
25	4	3	17	36	18	13	30
8	1	0	7	21	**71**	**95**	52
6	1	2	5	15	**92**	**114**	60
20	5	1	2	31	44	75	43

Figure 1: NM (Nested-Means) discretization. The numbers in the matrix shows the number of points that fall in each grid cell.

The EI approach divides a dimension into a number of equal-length intervals. Extreme outlier values can severely affect the effectiveness of the EI approach. The NM approach calculates the mean value of a dimension and divide the data into two halves with the mean value. Recursively, each half is divided again into halves with its own mean value. The process stops when the required number of intervals are obtained. The NM discretization can examine detailed structures within dense regions and, at the same time, can capture coarse patterns in comparatively sparse regions. NM approach is robust to outlier values.

The number of intervals (r) needed for each dimension depends on the data set size (n). A general rule adopted here is that on average each cell should contain about 35 points according to Cheng et al. (1999). Our experiments showed that following this rule can reliably achieve good results. Another rule is that, for the nested-means discretization, r should equal 2^k (k is a positive integer). Since we are processing 2-D spaces, $n/r^2 \approx 35$ and $r = 2^k$. For example, if $n = 10000$, then $r = 16$, because 16*16 = 256 and 256 *35 = 8960 (close to 10000). To scale well with extremely large data sets, the threshold 35 can increase by a factor of $\log_k n$, where k is a large integer (e.g., 1000). Thus the time complexity for discretizing all 2-D subspaces is $O(d^2 n \log n)$.

We are now experimenting and comparing other discretization techniques as well. From the users' perspective, it is also desirable to have a collection of discretization methods to interactively choose from.

3.2. Calculation of Conditional Entropy

The calculation of a conditional entropy given a matrix of values can be found in (Pyle 1999). We use the nested-means discretization instead of an equal-interval approach. Let S be a 2-D subspace comprising of dimensions a_i and a_j. To calculate the conditional entropy of S, both a_i and a_j. need to be discretized into ξ intervals. Thus S is partitioned into a matrix of grid cells. Let χ be the set of grid cells (including empty ones) for a column C in the matrix, and $d(x)$ be the density of a cell $x \in \chi$, i.e., the number of points in x divided by the total number of points in the column. Then the entropy of this column is calculated using the following equation:

$H(C) = - \sum_{x \in \chi} [d(x) \log d(x)] / \log|\chi|$.

Conditional entropy (Y|X) is a weighted sum of the entropy values of all columns (*figure 2*). Following are three steps to calculate conditional entropy (Y|X). Conditional entropy (X|Y) can be calculated similarly using rows instead of columns (see *figure 2*).

1) Calculate the column sum and weight (column sum divided by n—the data size).
2) Calculate the entropy for each column.
   For example, for column *x2*, $H(x2) = - [(1/36) * \log(1/36) + (9/36)*\log(9/36) + \ldots + (2/36)*\log(2/36)] / \log(6) = 0.847$, where 36 is the total number of points in column *x2* and 6 is the total number of cells in column *x2*.
3) Conditional entropy (Y|X) is a weighted sum of all column entropy values. It is 0.7 in figure 2.

	x1	x2	x3	x4	x5	x6		*Sum*	*Wt.*	*CE*
y1	0	1	3	0	0	0		4	.03	.314
y2	1	9	1	0	1	2		14	.09	.629
y3	7	14	3	7	6	0		37	.25	.835
y4	7	6	13	19	12	5		62	.41	.939
y5	0	4	14	5	1	1		25	.17	.668
y6	1	2	3	2	0	0		8	.05	.737
									CE(X|Y) .812	
Sum	16	36	37	33	20	8				
Wt.	.11	.24	.25	.22	.13	.05	CE(Y|X) .700		**CE*max* .812**	
CE	.597	.847	.806	.615	.540	.502				

Figure 2: The calculation of conditional entropy (Y|X) and conditional entropy (X|Y). The larger one of the two conditional entropy values is then taken as the final entropy value for the subspace.

As shown in figure 2, a 2-D subspace has two conditional entropy values (one for columns, *CE*(Y|X), and one for rows, *CE*(X|Y)). If both values are small, the two dimensions are highly correlated and the subspace has a strong cluster tendency. If one value is small, while the other is large, that means the data is only clustered well on one dimension and thus the two dimensions is not well correlated. Therefore, the maximum value, $CEmax = \max(CE(Y|X), CE(X|Y))$, is taken as the final cluster tendency measure of the subspace. The measure value will be only used for comparison between subspaces. Therefore the absolute value of the measure is not important although it does convey some information. Thus the measure is robust to noise if the noise exist in all subspaces.

4. SUBSPACE SELECTION USING CONDITIONAL ENTROPY MATRIX

Let $A = \{A_1, A_2, \ldots, A_d\}$ be a set of dimensions and $S = A_1 \times A_2 \times \ldots \times A_d$ be a d-dimensional data space. Let $S_2 = \{A_i \times A_j \mid i=1..d, j = 1..d, i < j\}$ be the set of all possible 2-D subspaces from this data space. The conditional entropy values of these 2-D subspaces can form a matrix. This matrix can also be viewed as a complete graph with each dimension as a vertex. There is an edge between any two dimensions. Since the range of those conditional entropy (CE) values is [0, 1], the length of each edge (or *distance* between two dimensions A_i and A_j) is assigned as (1 – CE(i, j)). This graph of dimensions will be used below for both visualizing the matrix and searching multidimensional subspaces that contain clusters.

The rationale for selecting multidimensional subspaces based on this matrix (graph) is*: if an L-dimensional (2 < L < d) subspace S_L has good clusters, all possible 2-D subspaces of S_L should have low conditional entropy values*. This rationale is similar to the monotonicity lemma used by CLIQUE: *if a collection of points P is a cluster in a k-dimensional space, then P is also part of a cluster in any (k-1)-dimensional projections of this space* (Agrawal, Gehrke et al. 1998). Is it possible that a subspace S_L has good clusters but some (or even all) of its projected 2-D subspaces do not have good clusters (and hence low conditional entropy values)? It is possible. However, since we need not to actually identify clusters, the overlap of clusters in a projected 2-D subspace is not a problem. Actually the overlap of clusters can make the conditional entropy value of the subspace even smaller. The only problematic case is that, all clusters are of the same density and are regularly distributed (no overlap and side by side) in the projected 2-D subspace. In such a case, the entropy value would be very high. However, the probability of such a case is very small.

4.1. Ordering Dimensions for Better Visualization

To render a better display of the entropy matrix, an optimal ordering of all dimensions is derived such that correlated dimensions (in terms of low conditional entropies) are placed as close to each other as possible in the ordering. The more highly correlated two dimensions are, the closer they should be in the ordering. A minimum spanning tree (MST) is constructed from the complete graph of all dimensions. From the MST, an optimal ordering of dimensions can be derived to completely preserve the hierarchical cluster structure and other proximity information of dimensions (Guo, Peuquet et al. 2002). Figure 3 shows the matrix with ordered dimensions.

Those dimensions of a multidimensional subspace with good clusters are likely to neighbor each other. Each cell is colored according to its entropy values—lower entropy values are assigned brighter colors. Striking bright blocks the indicate subspaces with good clusters. Both entropy values (bottom-left half) and correlation values (top-right half) are shown in the same matrix for comparison. But the correlation values are just used here for comparison—they are not used for searching multidimensional subspaces, which will be introduced next. The generation of the data set used here will be presented in section 5. From the comparison (in figures 3, 4, and 5) it can observed that those 2-D subspaces with a good correlation value always have a good conditional entropy value as well.

So far, the user has not yet given any subjective parameters, but he/she already gets a good overall understanding of the relationships among dimensions. The user can easily identify interesting subspaces that have good clusters based on the visual display. In figure 3 there are two interesting subspaces: {d9, d1, d3} and {d6, d2, d7, d8}. The data set used here only has 10 dimensions, which is designed for illustration only. Our approach can effectively and efficiently handle more than 100 dimensions and clusters of various dimensionalities. The time complexity for constructing the matrix is $O(d^2 n \log n)$, where d is the dimensionality and n is the size of the data set. Moreover, the construction of the matrix is decomposable. It can adopt a distributed

computing strategy to be time-efficient, or sequentially process data column by column (or row by row) to be memory-efficient.

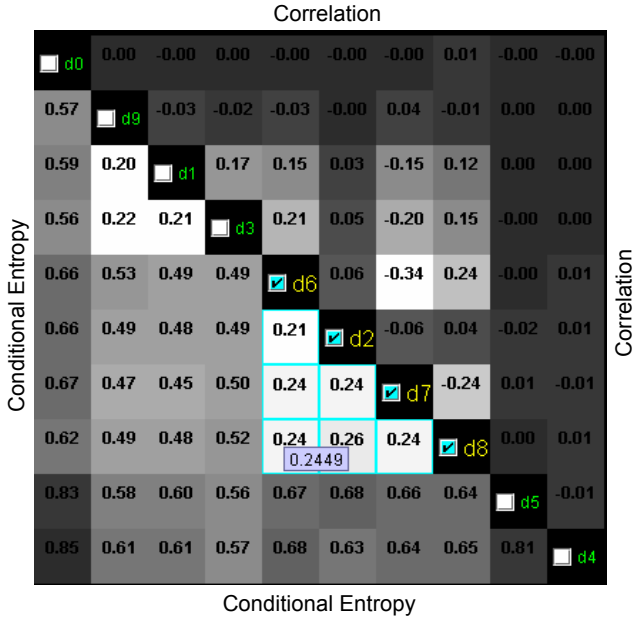


Figure 3: Visualization of the conditional entropy matrix. Dimensions are ordered according to a hierarchical clustering of those dimensions. Both entropy values (bottom-left half) and correlation values (top-right half) are shown for comparison.

Another advantage of this approach is that, once the matrix is constructed, the user can examine various relationships among dimensions without running the procedure again and again.

4.2. Automatic Search of Maximum Subspaces

Given a threshold of conditional entropy ***e***, a maximum subspace $S_{max}(e)$ satisfies two conditions: (**1**) the conditional entropy value of any 2-D subspace from $S_{max}(e)$ is smaller than e; and (**2**) adding any new dimension that is not in $S_{max}(e)$ will violate the first condition. As introduced above, if we regard the matrix as a complete graph, the search of maximum subspaces is similar to a complete-linkage clustering problem (Jain and Dubes 1988; Duda, Hart et al. 2001).

For a high-dimensional data set, it is desirable to have such an automatic procedure to efficiently identify those important subspaces. Figures 8 and 9 show the interface for this function. The user only needs to input a conditional entropy threshold (e), which is easy to set with the help of the visualization of the matrix. If the matrix is too big to label each value, then when the mouse is over a cell, its value pops out (see *figure 3*). The user can also interactively form a subspace according to his/her understanding, expertise, and interest.

5. EXPERIMENTS WITH SYNTHETIC DATA

We now evaluate our approach against a most recent subspace clustering method DOC (Procopiuc, Jones et al. 2002). We are still conducting experiments on various aspects, so only a preliminary result of the experiments is reported here. We used the data generator described in DOC to generate high-dimensional data sets that contain U-clusters, MGq-clusters, or N-clusters.

Our method works well with all these types of cluster, although it works better with U-clusters and MG*q*-clusters than with N-clusters. Here we present the results with N-clusters.

The dimensionality of data sets is 50 or 200. The data set size is 50,000 or 100,000. The *cluster dimensionality* (i.e., the dimensionality of the subspace associated with a cluster) ranges between 3 and 10 (from a Poisson distribution with a mean value of 5) in the first two experiments, and ranges between 35 and 45 (from a Poisson distribution with a mean value of 40) in the third experiment. This is designed to show that our approach can reliably identify both low-dimensional subspaces and high-dimensional subspaces. We also experimented with cases in which clusters share a fair portion of dimensions. There are always 5 clusters, which is the same as in DOC. The coverage of clusters ranges from 10% to 20%. The noise level is around 20%.

All generated points have values in the range [0, 100]. Clustered points are normally distributed in the associated subspace of bounded dimensions. The standard deviation of the clustered points for each dimension is randomly chosen from range [5, 10]. The mean value of the clustered points is also randomly chosen. The dimensionality of each cluster is also randomly chosen from range [3, 10] (or [35, 45] for the third experiment). Then the dimensions for each subspace are randomly chosen. For the second experiment, to simulate the situation where clusters share many dimensions, we force the selection of dimensions to focus on a small set of dimensions (e.g. 15).

Cluster ID	Dimensionality	Subspace dimensions	Points
1	10	{10,12, 13, 14, 19, 24, 25, 34, 48, 49}	8738
2	3	{31, 47, 38}	6188
3	4	{15, 20, 28, 44}	8688
4	4	{3, 11, 17, 19}	9335
5	7	{3, 9, 10, 26, 36, 40, 42}	7048
noise	50		10003
total	50		50,000

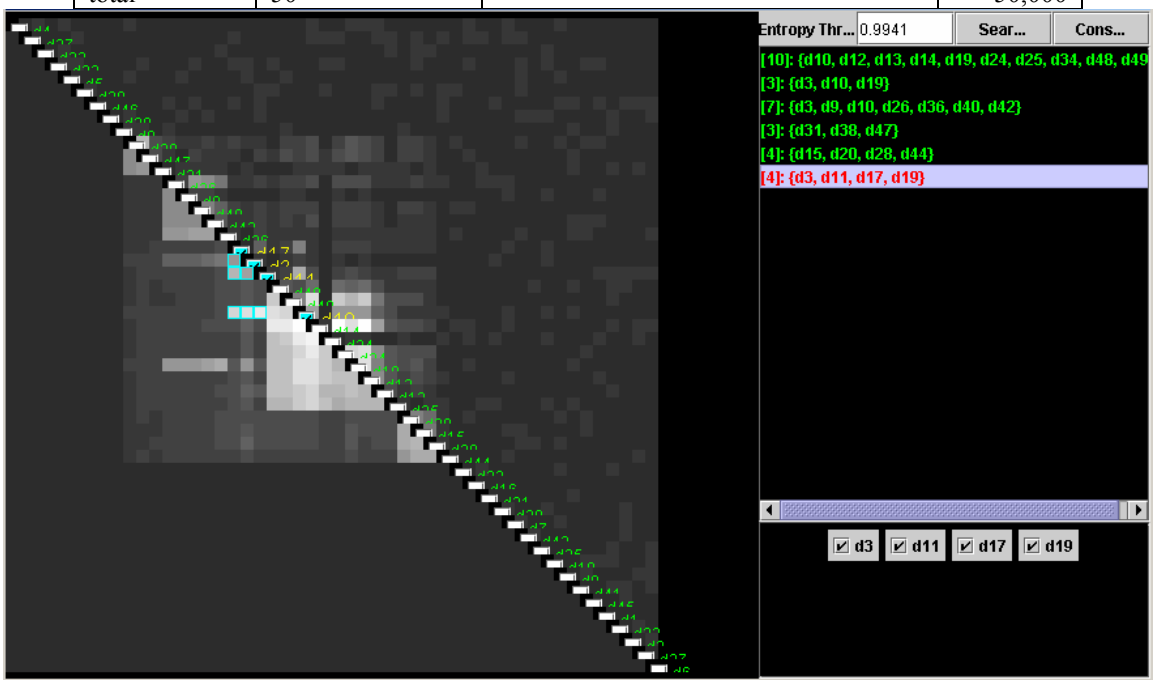


Figure 4: Experiment 1

Our approach is interactive in that it uses visualization to help users correctly select the only required threshold for automatic searching of subspaces. We have run the system many times

(each time it will generate a new data set). The following examples are common and representative. Figure 4 shows a simple case. There are 50 dimensions, 50000 points and 5 clusters. Clusters may share 1 or 2 dimensions with each other. From the matrix one can see that the difference between clustered and non-clustered dimensions is quite obvious. Dimensions of a subspace that has cluster(s) are likely to be ordered next to each other and form a bright block in the view. The threshold is easy to set and all 5 subspaces with clusters can be correctly identified.

Figure 5 shows the result of the overlapping case: clusters share many dimensions with each other. In following table, bold dimensions are shared by at least two clusters. In the snapshot, the view is zoomed to show those 15 dimensions (out of 50) that are involved in clusters. For this case, the threshold is a little difficult to set. Nevertheless, a proper threshold value can be found, with the help of visualization and human interaction, to identify those 5 heavily overlapped subspaces. However, due to the heavy overlap, one clustered subspace may be totally covered by another clustered subspace. For example, in figure 5, subspace{2, 5, 8, 12} is covered by subspace {1, 2, 5, 9, 8, 10, 12}, and therefore they cannot be separated. It can also be observed that subspace{3, 4, 5, 6, 8} is identified as {3, 4, 5, 6, 8, **2, 9, 10**}. It is because dimensions {2, 9, 10} are correlated with all dimensions in {3, 4, 5, 6, 8} via clusters 1 and 5 (see table below). Nevertheless, although the result subspace is not the exact one, it still uncovers important patterns.

Cluster ID	Dimensionality	Subspace dimensions	Points
1	7	{1, **2, 5, 9, 8, 10, 12**}	8030
2	5	{**3, 4, 5, 6, 8**}	7485
3	6	{**0, 2, 4,** 11, 13, **14**}	9559
4	4	{**2, 5, 8, 12**}	6852
5	9	{**0, 2, 3, 4, 6, 8, 9, 10, 14**}	7973
noise	50		10101
total	50		50,000

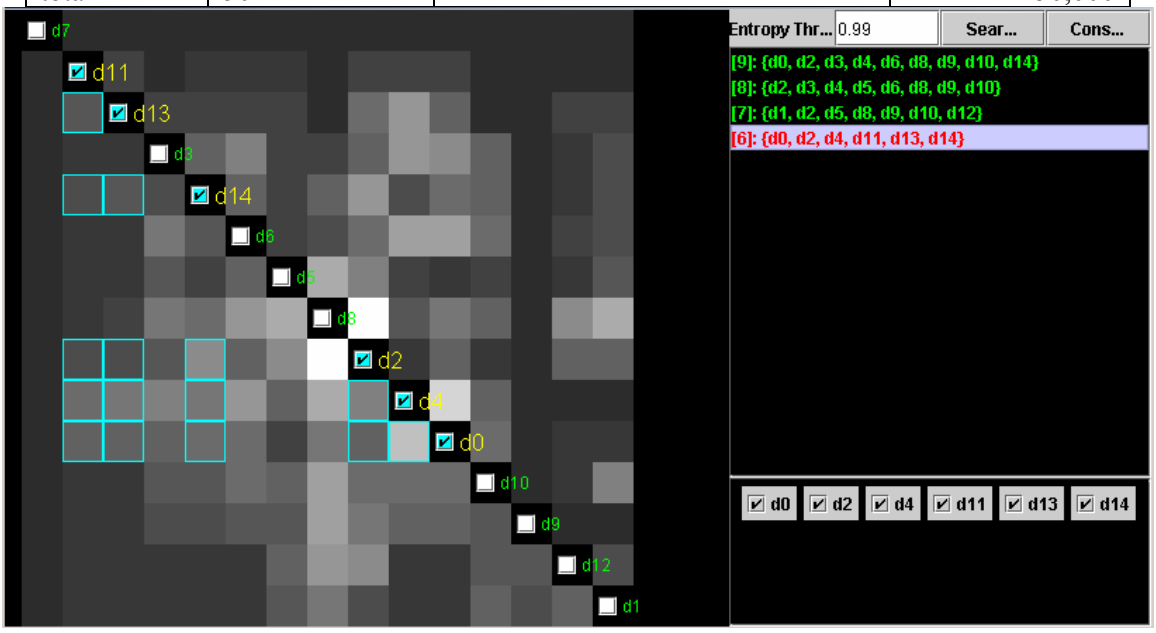


Figure 5: Experiment with clusters that share a large portion of dimensions. Bold numbers in the above table show dimensions that are at least shared by two clusters.

Figure 6 shows the third experiment with the case described in DOC. There are 200 dimensions, 100000 data points, and 5 clusters of dimensionality around 40. Our approach still works quite well for this case. By a careful examination with the visualization, a proper threshold

can be found to correctly identify all 5 clustered subspaces. Many low-dimensional subspaces are also produced due to the overlap effect (as introduce above). However, the result is still very satisfactory because 1) correct subspaces are all found, and 2) byproduct subspaces also reveal important associations among dimensions. Another desirable fact is that, if the threshold is set a little smaller, children subspaces of those clustered subspaces are found but no irrelevant dimensions included. If the threshold is a little too high, all the clustered subspaces are identified but may also include several more irrelevant dimensions.

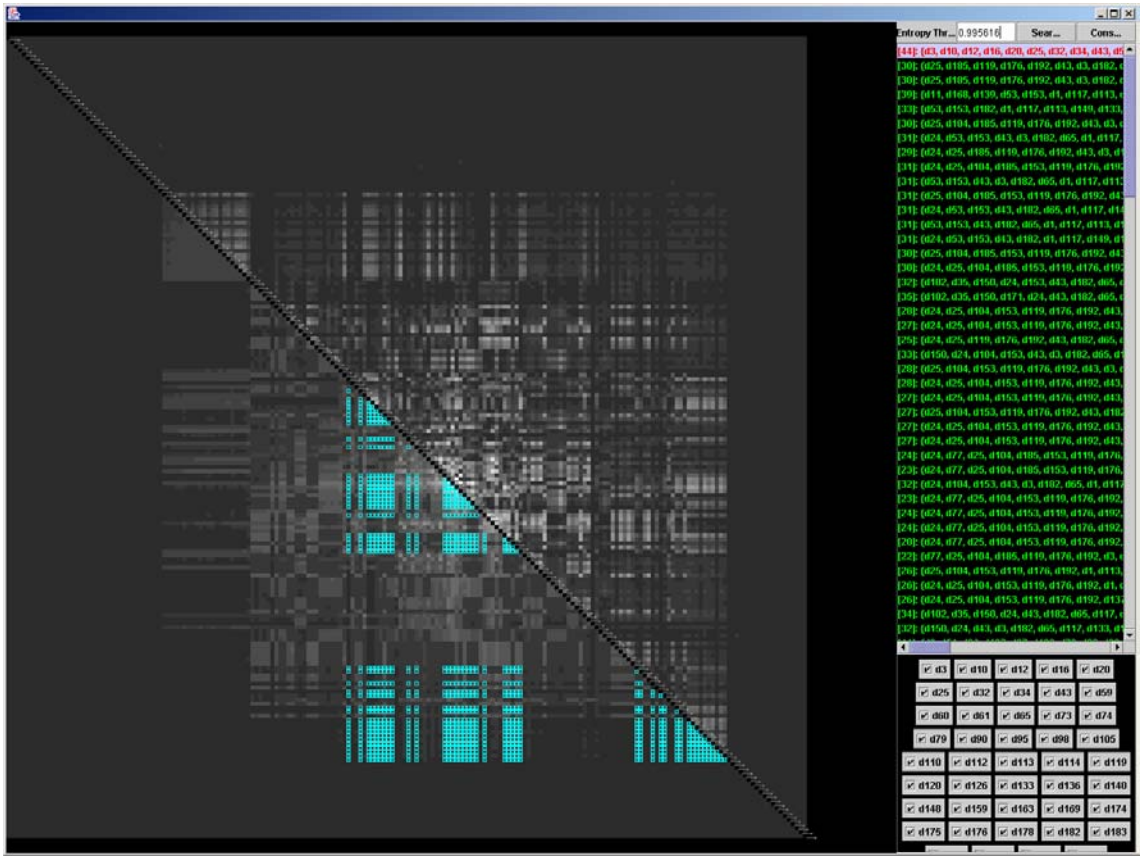


Figure 6: 200 dimensions and 5 clusters of dimensionality around 40.

6. CONCLUSION AND DISCUSSION

A subspace selection method is proposed based on a matrix of pair-wise conditional entropy values of 2-D data spaces. It does not need to identify the actually clusters. A new calculation of the conditional entropy is proposed to reliably measure the existence and the significance level of clusters (defined as a contiguous, dense region of arbitrary shape) in a 2-D space. From the entropy matrix, the user can easily get an understanding of the overall picture of various relationships among dimensions. Multidimensional subspaces of more than two dimensions can then be automatically extracted or interactively identified from the matrix. Experiments with synthetic data sets show that this approach is robust for correct identification of subspaces of various dimensionalities that contain clusters of various sizes. Other than choosing subspaces for clustering, this method can also be used to (1) inform various high-dimensional visualization techniques to focus on a subspace for better views; and (2) add, remove, and/or extract attributes to prepare a better data set for exploratory analysis (not limited to clustering).

This is an on-going research and there are several aspects that need careful examination. A more rigorous and complete evaluation of the approach is in progress.

Acknowledgement:

This paper is partly based upon work funded by NSF Digital Government grant (No. 9983445) and grant CA95949 from the National Cancer Institute.

References:

Agarwal, C., C. Procopiuc, J. Wolf, P. Yu and J. Park (1999). A Framework for Finding Projected Clusters in High Dimensional spaces. ACM SIGMOD International Conference on Management of Data.

Aggarwal, C. and P. Yu (2000). Finding generalized projected clusters in high dimensional spaces. ACM SIGMOD International Conference on Management of Data.

Agrawal, R., J. Gehrke, D. Gunopulos and P. Raghavan (1998). Automatic subspace clustering of high dimensional data for data mining applications. ACM SIGMOD International Conference on Management of Data, Seattle, WA USA.

Alsabti, K. and V. S. Sanjay Ranka (1998). An Efficient K-Means Clustering Algorithm. IPPS: 11th International Parallel Processing Symposium.

Bradley, P. and U. Fayyad (1999). "Efficient Probabilistic Data Clustering: Scaling to Large Databases."

Bradley, P., U. Fayyad and C. Reina (1998). Scaling clustering algorithms to large databases. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York City.

Cheng, C., A. Fu and Y. Zhang (1999). Entropy-based subspace clustering for mining numerical data. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA.

Cheng, C., A. Fu and Y. Zhang (1999). Entropy-based subspace clustering for mining numerical data. Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Duda, R. O., P. E. Hart and D. G. Stork. (2001). Pattern classification. New York, John Wiley & Sons.

Dy, J. G. and C. E. Brodley (2000). Feature subset selection and order identification for unsuprervised learning. the Seventeenth International Conference on Machine Learning, Stanford University.

Dy, J. G. and C. E. Brodley (2000). Visualization and interactive feature selection for unsupervised data. the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, Boston, Massachusetts, United States, ACM Press New York, NY, USA.

Ester, M., H.-P. Kriegel, J. Sander and X. Xu (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. the 2nd International Conference on Knowledge Discovery and Data Mining, Portland, Oregon, AAAI Press.

Fayyad, U., G. Piatetsky-Shapiro and P. Smyth (1996). From data mining to knowledge discovery-An review. Advances in knowledge discovery. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusay. Cambridge, MA, AAAI Press/The MIT Press**:** 1-33.

Guha, S., R. Rastogi and K. Shim (1998). CURE: An Efficient Clustering Algorithm for Large Databases. ACM SIGMOD International Conference on Management of Data, Seattle, Washington.

Guo, D., D. Peuquet and M. Gahegan (2002). Opening the Black Box: Interactive Hierarchical Clustering for Multivariate Spatial Patterns. The 10th ACM International Symposium on Advances in Geographic Information Systems, McLean, VA, USA.

Hinneburg, A. and D. A. Keim (1999). Optimal Grid-Clustering: Towards Breaking the Curse of Dimensionality in High-Dimensional Clustering. Proceedings of the 25th VLDB Conference, Edingburgh, Scotland.

Jain, A. K. and R. C. Dubes (1988). Algorithms for clustering data. Englewood Cliffs, NJ, Prentice Hall.

Jain, A. K., M. N. Murty and P. J. Flynn (1999). "Data clustering: a review." ACM Computing Surveys (CSUR) **31**(3): 264 - 323.

Kim, Y., W. N. Street and F. Menczer (2000). Feature selection in unsupervised learning via evolutionary search. Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, Boston, Massachusetts, United States, ACM Press New York, NY, USA.

Liu, H. and H. Motoda (1998). Feature selection for knowledge discovery and data mining. Boston, Kluwer Academic Publishers.

Pelleg, D. and A. Moore (1998). "Accelerating Exact k-means Algorithms with Geometric Reasoning."

Procopiuc, C. M., M. Jones, P. K. Agarwal and T. M. Murali (2002). A Monte Carlo Algorithm for Fast Projective Clustering. ACM SIGMOD, Madison, Wisconsin, USA, ACM.

Pyle, D. (1999). Data preparation for data mining. San Francisco, Calif., Morgan Kaufmann Publishers.

Schikuta, E. (1996). Grid clustering: An efficient hierachical clustering method for very large data sets. 13th Conf. on Pattern Recognition, IEEE Computer Society Press.

Slocum, T. A. (1999). Thematic cartography and visualization, Upper Saddle River, N.J. : Prentice Hall.

Snedecor, G. W. and W. G. Cochran (1989). Statistical methods, Iowa State University Press.

Zhang, C. and Y. Murayama (2000). "Testing local spatial autocorrelation using k-order neighbors." International Journal of Geographical Information Science **14**(7): 681-692.

Zhang, T., R. Ramakrishnan and M. Livny (1996). BIRCH: An Efficient Data Clustering Method for Very Large Databases. ACM SIGMOD Int. Conf. on Management of Data, Montreal, Canada, ACM Press.