Information Visualization (2003) 2, 232–246
© 2003 Palgrave Macmillan Ltd. All rights reserved 1473-8716 $25.00
www.palgrave-journals.com/ivs

Coordinating computational and visual approaches for interactive feature selection and multivariate clustering

Diansheng Guo[1]

[1]GeoVISTA Center & Department of Geography, The Pennsylvania State University, University Park, PA, U.S.A.

Correspondence:
Diansheng Guo, GeoVISTA Center & Department of Geography, The Pennsylvania State University, 302 Walker Building, University Park, PA 16802, U.S.A.
Tel: +1 814 865 3433;
fax: +1 814 863 7943;
E-mail: dguo@psu.edu

Received: 05 August 2003
Revised: 15 September 2003
Accepted: 17 September 2003

Abstract

Unknown (and unexpected) multivariate patterns lurking in high-dimensional datasets are often very hard to find. This paper describes a human-centered exploration environment, which incorporates a coordinated suite of computational and visualization methods to explore high-dimensional data for uncovering patterns in multivariate spaces. Specifically, it includes: (1) an interactive feature selection method for identifying potentially interesting, multidimensional subspaces from a high-dimensional data space, (2) an interactive, hierarchical clustering method for searching multivariate clusters of arbitrary shape, and (3) a suite of coordinated visualization and computational components centered around the above two methods to facilitate a human-led exploration. The implemented system is used to analyze a cancer dataset and shows that it is efficient and effective for discovering unknown and unexpected multivariate patterns from high-dimensional data.

Information Visualization (2003) **2,** 232–246. doi:10.1057/palgrave.ivs.9500053

Keywords: data mining and knowledge discovery; feature selection; mutual information; entropy; interactive visualization; hierarchical clustering

Introduction

Owing to the increasing size of datasets and the increasing complexity of problems to be addressed, datasets to be analyzed nowadays often have many, for example, more than 50, dimensions (or variables) and sometime are very large – often having more than 10,000 observations. Such datasets are commonly compiled from multiple data sources, which might have also been collected for varying purposes. By putting them together for analysis, we are hoping to find unknown (and unexpected) complex relationships or patterns that may involve multiple variables.

The high dimensionality of such datasets can cause serious problems for almost all data analysis methods, especially for unsupervised, exploratory approaches (including both automatic algorithms and interactive visualization techniques). The quality and relevance of variables can vary dramatically. It is important for a data analysis approach employed to be able to discriminate relevant dimensions from irrelevant dimensions and include only the former in further analysis. Otherwise, irrelevant dimensions may hide rather than help uncover relationships or patterns. Most existing data analysis methods cannot discriminate relevant variables from irrelevant ones and rely on the user (presumed to be an expert on the application problem) to provide a meaningful set of variables for further analysis. Moreover, relationships or patterns may exist in different *subspaces*,[1] that is, different relationships may involve different subsets of the original dimensions and these subsets may or may not overlap with each other.[1–3]

Depending on the user to choose variables according to her/his expertise or hypothesis makes it impossible to find *unexpected* patterns, while finding such unexpected patterns is one of the main purposes of data mining and knowledge discovery.[4] Moreover, given the high dimensionality of currently available data and the huge set of possible hypotheses, such a manual approach for feature selection is at best inefficient and at worst humanly impossible.

Feature selection methods have been studied in the area of supervised classification.[5] Recently, several unsupervised feature selection methods have been developed to select an 'optimal' subset of dimensions,[6,7] or produce a pool of dimension subsets,[8] for unsupervised clustering. Since clusters may exist in different subspaces, it can be ineffective or impossible to find a single 'optimal' subset of dimensions for identifying all clusters.[9] To search clusters in different subspaces, several subspace clustering methods have been developed.[1–3,10]

Nevertheless, it remains a challenging research problem to effectively identify interesting subspaces from a high-dimensional data space and then search patterns in each of them. The difficulties in addressing this problem are twofold. On one hand, confronting the large number of dimensions and the consequent combinatory explosion of possible subspaces, automatic computational methods must play an important role. On the other hand, considering various forms that unknown patterns may take and the interpretation and evaluation of each discovered pattern, human expertise and interaction is indispensable and needed to guide the computational procedure.

To achieve both efficiency and effectiveness for exploring high-dimensional (and sometime large) datasets, research cannot focus on either computational methods or visualization techniques in isolation.[11] A powerful data mining strategy lies in tightly coupling visualization techniques and analytical processes into a unified framework.[12] It is critical to coordinate computational methods and visualization techniques to integrate the best of both human and machine capabilities.[13]

The research reported upon here develops a human-centered and component-oriented knowledge discovery environment for exploring large and high-dimensional data. The implemented prototype system includes a suite of computational and visualization methods, each of which focuses on a specific task or step in the overall data exploration process and together they can communicate with each other and collaboratively address complex problems. Specifically, the research includes: (1) an interactive feature selection method for identifying interesting subspaces, (2) an interactive, hierarchical clustering method for searching arbitrary-shaped multivariate clusters, and (3) a suite of coordinated visualization and computational components centered around the above two methods to facilitate an efficient and effective human-led exploration of high-dimensional and large data.

The remainder of the paper is organized as follows. The next section gives a review of related research. The section following the next presents the interactive feature selection method developed. Later, the interactive, hierarchical, multidimensional clustering method employed are introduced. In the section penultimate, a component-based implementation of the system is presented. Finally, the application of the developed methods and system in analyzing a cancer dataset is presented.

Related work

A traditional way to address high dimensionality is to apply a dimension reduction method to the dataset. Such methods include principle component analysis (PCA) and self-organizing maps (SOM),[14] which transform the original data space into a low-dimensional space. While these techniques may succeed in reducing dimensionality, they fall short in exploring various subspace patterns from high-dimensional data. PCA is not effective in identifying relationships or patterns that reside in different subspaces (see Procopiuc *et al.*[2] for detailed explanation). SOM uses measurements from all original dimensions to derive the projection to a 2-D space and therefore noisy or irrelevant dimensions have dramatic impacts on the projection result.

Feature selection methods are traditionally used to select a subset of dimensions for supervised classification problems.[5] Recently, several unsupervised feature selection methods have been developed to select either an 'optimal' subset of features for unsupervised clustering,[6,7] or produce a pool of 'good' dimension subsets for searching clusters.[8] Each of these methods centers around a specific clustering method, for example the expectation maximization[6] or the K-means.[8] 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 therefore a feature selection method based on a K-means method may be biased toward dimension subsets that contain circular clusters.

To enhance the detection of multivariate patterns in high-dimensional data, sorting variables has been an important step for visualizing high-dimensional dataset. The idea is to place correlated or similar dimensions close to each other in the high-dimensional visual space to help the human user perceive relationships among these vaiables. To arrange dimensions for visualizing a correlation matrix, an ordering of variables based on bivariate linear correlation values is developed in.[15] To help detect sequential patterns in high-dimensional visualization, a similarity-based approach to sorting variables is developed in.[16]

Clustering analysis organizes a set of objects into groups (or clusters) such that objects in the same group are similar to each other and different from those in other groups.[17,18] Clustering methods can be divided into two types: *partitioning* and *hierarchical* approaches. The partitioning approach aims to divide the dataset into several clusters, which may not overlap with each other but together cover the whole data space. Hierarchical clustering approaches decompose the dataset with a sequence of nested partitions, from fine to coarse resolution. Hierarchical clustering can be presented with dendrograms, which consist of layers of nodes, each representing a cluster.[19] In each group (i.e., partitioning or hierarchical), the methods can be further classified into three subgroups: distance-based, model-based, and density-based. See Jain *et al.*[20] and Guo *et al.*[21] for a detailed review on clustering methods.

Density-based clustering methods have been developed mainly for dealing with large datasets and identifying arbitrary-shaped clusters. Such methods regard a cluster as a dense region of data objects.[1,19,22] Density-based clustering can adopt either a grid-based or a neighborhood-based approach. A grid-based approach divides the data space into a finite set of multidimensional grid cells, calculates the density of each grid cell, and then groups those neighboring dense cells into a cluster. Such methods include Grid-Clustering,[23] CLIQUE,[1] OptiGrid,[24] ENCLUS.[3] For neighborhood-based approaches, the neighborhood – either a hyper-sphere of radius ε (as in DBSCAN[22] and OPTICS[25]) or a hyper-cube of side length w (as in DOC[2]) – of an object has to contain at least a minimum number of objects (*MinPts*) to form a cluster around this object.

The integration of clustering methods and visualization tools has been explored by several researches. For example, a visual data mining system is developed in[26] to combine a clustering algorithm with visualization methods for interactive clustering of data. Presented in[27] is a hierarchical clustering explorer that allows users to control the processes and interact with the results based on traditional dendrograms.

Interactive feature selection

A new feature selection approach is developed to identify interesting subspaces (i.e., subsets of dimensions) from a high-dimensional space that potentially contain meaningful patterns. The developed feature selection method examines all 2-D subspaces of the original high-dimensional data space. The computational complexity of the method is $O(d^2 n \log n)$, where d is the total number of dimensions in the data and n is the total number of observations (or instances). The method first calculates a measure to evaluate the 'goodness of clustering' in a 2-D data space. Then it uses a hierarchical clustering approach to derive a sorting of all dimensions and produces an enhanced visualization to show relationships among dimensions. Interesting multidimensional subspaces consisting of more than two dimensions can then be identified, interactively or automatically. A detailed evaluation of this feature selection method with synthetic datasets (which contain controlled patterns) is presented in.[28] Here an extended and sophisticated version of the method, with various ancillary visualization features, is presented below.

In this paper, patterns or relationships specifically refer to various types of clusters, which are defined as contiguous, arbitrary-shaped, dense areas of data objects. A partition of the data space is not required, that is, a data space may contain only one cluster. So, a linear relationship can be regarded as a special case of a cluster, which has an elongated shape.

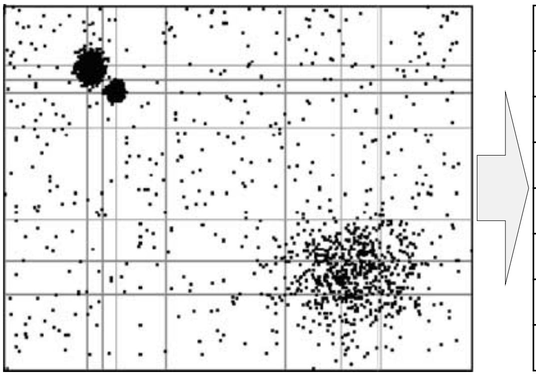
A measure of the 'goodness of clustering'

To measure the mutual information (or 'goodness of clustering') between two dimensions, there are three criteria to consider: high coverage (percentage of data points contained in clusters), high density (high coverage in a small region), and high dependence.[3] Different subspaces can have different number of clusters of various shapes, sizes, and distributions. Noise and extreme values can also exist. Thus, a suitable measure of the 'goodness of clustering' should not be biased towards any particular type of clusters and should be robust with extreme values and noise.

A calculation of *maximum conditional entropy* (MCE) is developed to measure the 'goodness of clustering' in a 2-D data space. A *critical* step in the calculation of an MCE value is to discretize the 2-D data space into a matrix of grid cells by cutting each dimension into a set of intervals. A nested-means (NM) discretization method is adopted. The advantage of an NM method over an equal-interval discretization method is discussed in.[21,28] The NM approach first calculates the mean value of a dimension and then divides the data into two halves with that mean value. Recursively, each half is divided into halves with its own mean value (Figure 1). The recursion stops when the required number of intervals is obtained.

The number of intervals (r) needed for each dimension depends on the dataset size (n). A general rule adopted here is that on average each 2-D cell should contain about 35 points according to Cheng *et al.*[3] Another rule is that, for the nested-means discretization, r should equal 2^k (k is a positive integer). For example, if $n = 10{,}000$, then $r = 16 = 2^4$, because $16^2 = 256$ and $256 * 35 = 8960$ (close to 10,000). To scale well with extremely large datasets, the threshold 35 can increase by a factor of $\log_k n$, where k is a large integer (e.g., 1000). Thus, the computational complexity for discretizing all 2-D subspaces in a d-dimensional dataset is $O(d^2 n \log n)$.

The calculation of a conditional entropy can be found in.[29] Let S be a 2-D subspace comprising of dimensions a_i and a_j. Both a_i and a_j are first discretized into ξ intervals using the NM method. 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 $x \in \chi$, that is, the number of points in x divided by the total number of points in the column.



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 Nested-means (NM) discretization of a 2-D data space. Each dimension is recursively cut into eight intervals. The number in each grid cell shows the number of points that fall in the cell.

	x1	x2	x3	x4	x5	x6		*Sum*	*Wt.*	*H(R)*
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		**MCE** **0.812**	
H(C)	.597	.847	.806	.615	.540	.502				

Figure 2 The calculation of conditional entropy (*Y*|*X*) and conditional entropy (*X*|*Y*) given a matrix of values. The larger one of the two conditional entropy values is then taken as the final entropy value (maximum conditional entropy – MCE) for the subspace.

Then the entropy of this column is calculated using the following equation:

$$H(C) = -\sum_{x\in\chi}[d(x)\log d(x)]/\log|\chi|. \qquad (1)$$

Conditional entropy (Y|X) is a weighted sum of the entropy values of all columns (Figure 2). Similarly, *conditional entropy (X|Y)* can be calculated using rows instead of columns (Figure 2). The *maximum conditional entropy* (*MCE*) value of this 2-D subspace is the maximum value of the above two conditional entropy values (Figure 2). The more clustered a dataset is in a 2-D space, the smaller its *MCE* value is. The *MCE* values are not used for testing statistical significance here. Rather, they are used for comparison. In other words, for variables a_i, a_j, a_m, and a_k in a dataset, if $MCE(a_i, a_j) < MCE(a_m, a_k)$, then we would say a_i and a_j have better clusters than a_m and a_k do.

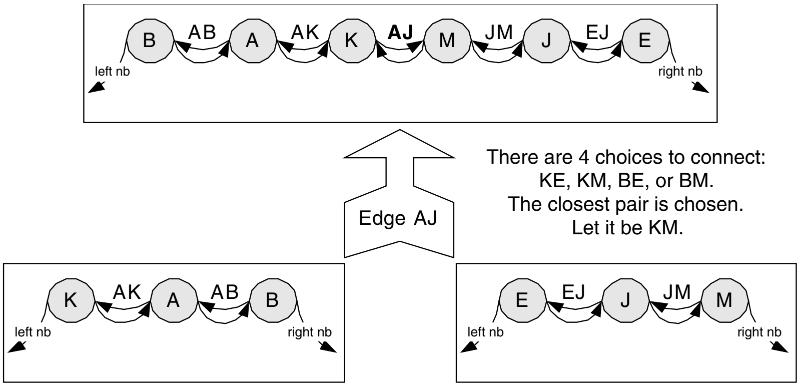
χ^2 test has been widely used for testing the statistical significance of bivariate tabular association, especially for nominal (categorical) data.[30] Since the 2-D data space is now discretized, a (χ^2) value of this 2-D space is also calculated for comparison with the maximum conditional entropy value (see next section).

Sorting dimensions for better visualization

Let $A = \{a_1, a_2, \cdots, a_d\}$ be a set of dimensions and $S = \mathrm{a}_1 \times \mathrm{a}_2 \times \cdots \times \mathrm{a}_d$ be a *d*-dimensional data space. Let $S_2 = \{\mathrm{a}_i \times \mathrm{a}_j | i = 1..\mathrm{d},\ j = 1..d,\ i < j\}$ be the set of all possible 2-D subspaces in *S*. The *MCE* values of those 2-D subspaces in S_2 form a symmetric matrix (hereafter *entropy matrix*). This entropy matrix can be viewed as a complete graph with each dimension as a vertex. Each *MCE* value can then be viewed as the distance (or dissimilarity) between two dimensions. Thus it can be imagined that there is an 'edge' between any two dimensions and the length of the edge is the *MCE* value between the two dimensions.

To render a better display of the entropy matrix, an ordering (or sorting) of all dimensions is needed such that correlated (in terms of a low *MCE* value) dimensions are placed as close to each other as possible in the ordering. The more correlated two dimensions are, the closer they should be in the ordering. As introduced in the section on 'Related work', there are several existing methods for sorting variables in high dimensional.[15,16] Here a new sorting approach is developed based a hierarchical clustering method, which is introduced below.

A minimum spanning tree (MST) is constructed from the complete graph of all dimensions depicted above. During the construction of the MST, a *unique* ordering of all dimensions can be achieved. At the very beginning of constructing the MST, each cluster contains a single point (dimension). When an edge is added into the MST, the edge will connect two clusters of dimensions into one. A cluster (connected graph) of dimensions can be viewed as a chain of 'points'.[31] Each chain has two end points (at the very beginning they are the same point). There are



There are 4 choices to connect:
KE, KM, BE, or BM.
The closest pair is chosen.
Let it be KM.

Figure 3 Deriving an ordering of dimensions from their entropy matrix with an MST-based approach. Adding edge AJ to the MST connects two chains of points – the closest pair of end points is connected (here it assumes edge KM is shorter than KE, BE, or BM).

four choices to connect two chains (see Figure 3). The closest two ends should be connected. Once all points (dimensions) are in the same chain, an ordering of dimensions is achieved.

Many other graph-based hierarchical clustering methods, for example, average-link and complete link methods,[19,32] can also be adopted here to derive a cluster hierarchy of dimensions, as long as there is a strategy to determine an 'optimal' connection when merging two clusters (chains) of dimensions into one.

Figure 4 shows the entropy matrix of a real dataset with 72 dimensions. This visualization of matrix can accommodate a large number of dimensions. Each cell with a color represents a measure value between two dimensions. MCE values of paired variables are displayed below the diagonal and χ^2 values of paired variables are displayed above the diagonal. In both cases, the brighter cells represent *good* values: low MCE values or high χ^2 values. With the mouse over a cell, the *MCE* (or χ^2) value of that cell will pop out. The diagonal provides access to each variable; the user can select, add to, or subtract from a subset by simply clicking on the variable's diagonal cell. A selected subset can be broadcast to other components (sorted on the conditional entropy values for the subspace selected) for further analysis.

From the matrix it can be observed that the MCE value and χ^2 value of a 2-D subspace agrees with each other very well. Note: both MCE values and χ^2 values are calculated by discretizing the 2-D data space with a nested-means approach (not equal-interval approach). The time complexity for constructing the entropy matrix is $O(d^2 n \log n)$, where d is the dimensionality and n is the size of the dataset. The most time-consuming part is the discretization of all 2-D subspaces. Once the entropy matrix is constructed and visualized, the user can examine various relationships among dimensions without running the procedure repeatedly.

Interactive exploration and interpretation of subspaces

With the entropy matrix, the user can get a holistic understanding of the relationships among dimensions. The user can identify potentially interesting subspaces that might have good clusters based on the visual display. For the complex, real data shown in Figure 4, one can easily perceive those 'hot spots' (blocks of bright colors), which are multidimensional subspaces that likely contain significant patterns.

The user can interactively form a subspace according to his/her understanding, expertise, and interest. For a large number of dimensions in real dataset, the relationships among dimensions can be very complex. Some dimensions may be duplicate and identical, which definitely will show strong patterns but are not so interesting to the user. Some patterns may not be as strong as the above ones but can be of great interest to the user. With the user's interactive exploration and interpretation with the matrix, meaningful and interesting subspaces can quickly be identified.

An automatic algorithm is also developed to help the user quickly locate potentially interesting subspaces. Given a threshold MCE value e, a *maximum subspace* $S_{\max}$ (e) satisfies two conditions: (1) the MCE value of any 2-D subspace from $S_{\max}$ (e) is lower than e; and (2) adding any new dimension that is not in $S_{\max}$ (e) will violate the first condition. An algorithm is developed to automatically search maximum subspaces given a threshold. Figure 5 shows a list of maximum subspaces identified from the entropy matrix shown in Figure 4. The user can select a subspace from this list and modify it by adding/removing variables according to the entropy matrix and her/his expertise and interest.

Interactive multivariate clustering

Once an interesting subspace is formed in the feature selection component, the next step is to search multivariate patterns in this subspace. Normally, the dimensionality of the selected subspace is much lower than the original data space. Many existing clustering methods or visualization techniques can be employed here to search patterns in this subspace. Since the goal of this research is to help the user *interactively* search multivariate clusters of arbitrary shape in large datasets, a

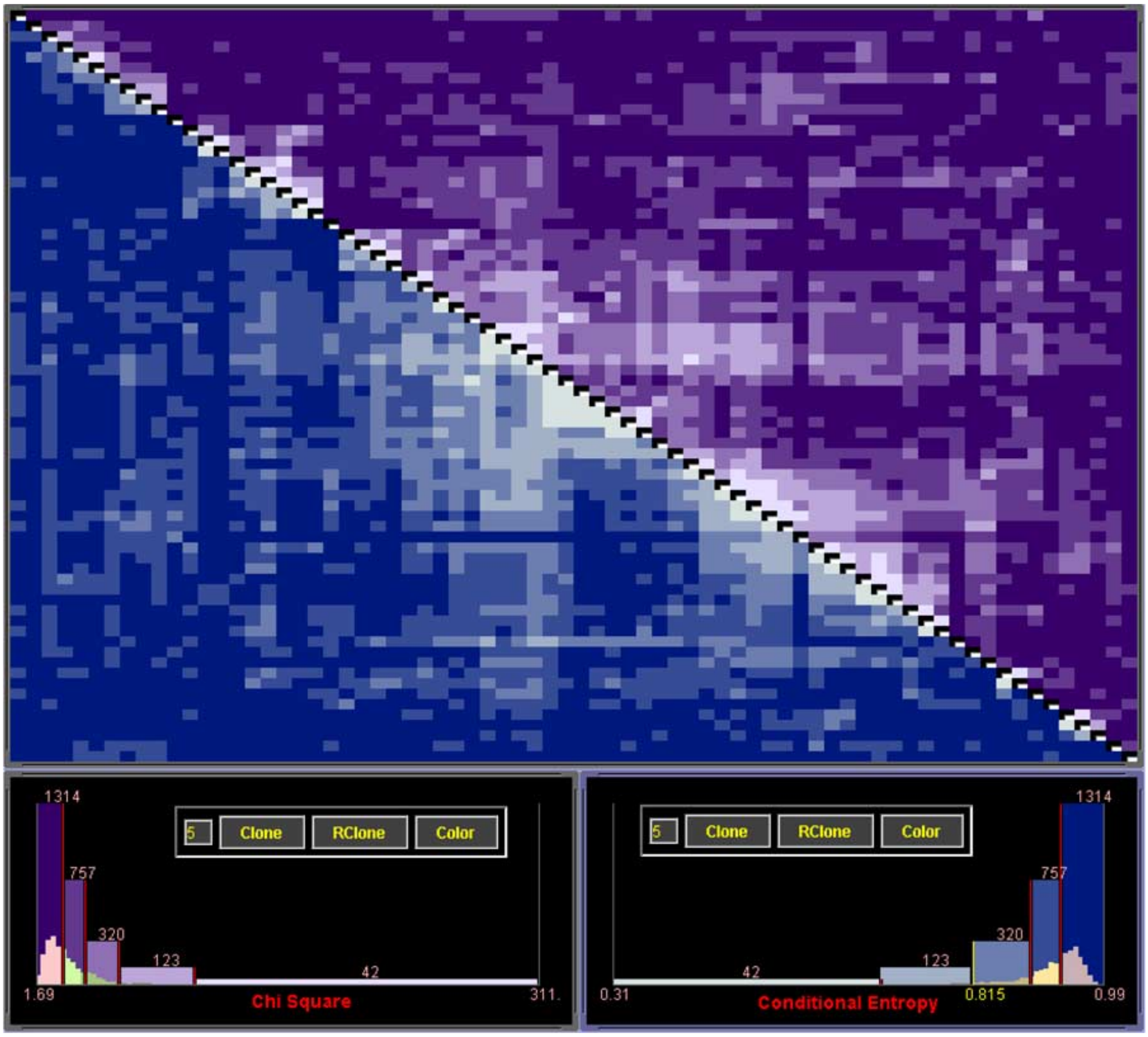
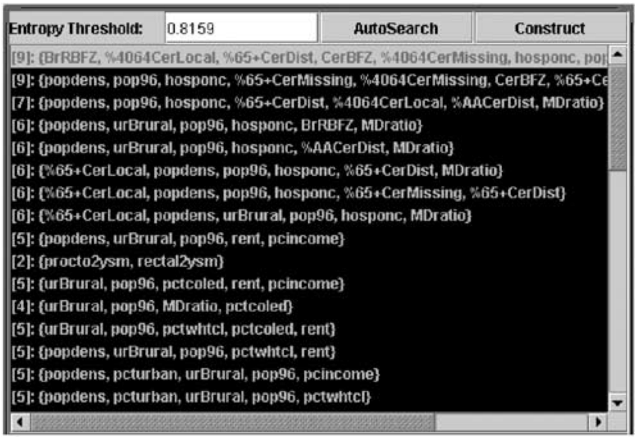


Figure 4 Visualizing the entropy matrix of a real cancer dataset with 72 dimensions. Two interactive histograms are implemented to flexibly adjust the classification and coloring of values.



```
Entropy Threshold: 0.8159
[9]: {BrRBFZ, %4064CerLocal, %65+CerDist, CerBFZ, %4064CerMissing, hosponc, po
[9]: {popdens, pop96, hosponc, %65+CerMissing, %4064CerMissing, CerBFZ, %65+Ce
[7]: {popdens, pop96, hosponc, %65+CerDist, %4064CerLocal, %AACerDist, MDratio}
[6]: {popdens, urBrural, pop96, hosponc, BrRBFZ, MDratio}
[6]: {popdens, urBrural, pop96, hosponc, %AACerDist, MDratio}
[6]: {%65+CerLocal, popdens, pop96, hosponc, %65+CerDist, MDratio}
[6]: {%65+CerLocal, popdens, pop96, hosponc, %65+CerMissing, %65+CerDist}
[6]: {%65+CerLocal, popdens, urBrural, pop96, hosponc, MDratio}
[5]: {popdens, urBrural, pop96, rent, pcincome}
[2]: {procto2ysm, rectal2ysm}
[5]: {urBrural, pop96, pctcoled, rent, pcincome}
[4]: {urBrural, pop96, MDratio, pctcoled}
[5]: {urBrural, pop96, pctwhtcl, pctcoled, rent}
[5]: {popdens, urBrural, pop96, pctwhtcl, rent}
[5]: {popdens, pcturban, urBrural, pop96, pcincome}
[5]: {popdens, pcturban, urBrural, pop96, pctwhtcl}
```

Figure 5 The automatic subspace searcher and a list of identified maximum subspaces.

density- and graph-based method is developed for hierarchical multivariate clustering. The initial idea of this method is presented in[21] but the subspace evaluation part of the original method is removed and replaced with the feature selection method introduced above. The focus here is on the integration and collaboration of this clustering method with other visualization components.

The clustering method first aggregates data points into a small number of non-empty hyper-cells, each of which may contain one or many data points. The method then extracts dense cells (given a density threshold that is interactively configured by the user) and thus can remove noise and makes major patterns easy to emerge. The clustering method is coupled with several interactive visualization interfaces to support interactive configuration of algorithm parameters and human-led exploration and interpretation of clusters.

The clustering method can enhance high-dimensional visualization techniques in several aspects. First, visualization components only need to visualize those dense cells (each of which as a summary 'point') rather than all original data points. Since the number of dense cells is much smaller than the original data size, it makes visualization methods more efficient, less clumpy, and clearer in presenting major patterns. Second, the clustering method can generate a 1-D ordering, which is a complete representation of the hierarchical cluster structure. This ordering makes it possible to use color continuum to represent hierarchical clusters and communicate among components.

Aggregation of data and extraction of dense cells

The input subspace is first partitioned into multidimensional cells. Such a discretization of a multidimensional subspace is similar to the discretization of 2-D space introduced in the previous section. Each dimension is cut into a set of intervals using the nested-means discretization. The number of intervals needed for each dimension is determined by the subspace dimensionality and the dataset size.

To further improve the effectiveness and efficiency of the clustering process, only *dense* cells are selected for further clustering. The density (or coverage) of a hyper cell is defined as the percentage of total points that is contained in the cell. A cell is dense if its density exceeds a density threshold set by the user. The density threshold is configured according to the distribution of the densities of all cells. A visual tool is developed to assist the user to interactively configure the threshold (see Figures 7, 9, and 11).

A proximity measure between dense hyper-cells

To find hierarchical clusters with a set of dense cells, a 'distance' measure is needed to define the similarity (or proximity) between two cells. Two facts need to be considered in deriving a distance between two cells: (1) the size of two cells can be different, and (2) the distribution of data points in each cell can be quite different. Here a synthetic distance measure is used,[21,33] which considers both the nominal position of intervals and the distribution of data points within each cell.

A synthetic value (*SynVal*) is calculated for each interval within a cell based on: the nominal position (*i*) of the interval on the dimension, the interval bounding values [Min_i, Max_i], and the mean dimension value ($Mean_i$) of all data points contained in the cell. Following is the equation for calculating the synthetic value of an interval:

$$SynVal = [(Mean_i - (Max_i + Min_i)/2)/(Max_i - Min_i)] + i. \tag{2}$$

Thus each cell has a vector of synthetic values, with which a cell is defined as a 'point' in the multidimensional space. A metric can then be used to derive a distance between two cells. Here the Euclidean metric is used (note: the Euclidean distance is calculated with those synthetic values of each cell). The most prominent characteristic of this proximity measure is that two cells are closer if the distribution of data points in each cell is more skewed towards each other. The developed method is designed and in future will be able to flexibly support a collection of distance measures for the user to choose and compare.

Hierarchical clustering and ordering of dense cells

With a set of dense cells and a proximity measure between cells, a matrix of pair-wise distance measures for all dense cells can be derived. This matrix is a complete graph with each dense cell as a vertex. A hyper-MST can then be derived. The algorithm for deriving a hyper-MST and an ordering of all dense cells is similar to the one for sorting dimensions introduced in the section on 'Interactive feature selection'. Since here only dense cells are considered (and sparse cells are excluded) in the construction of a hyper-MST, it can effectively avoid the single-link effect (or chaining effect). The computational complexity of this clustering method is $O(c^2)$, where c is the number of dense cells. Normally, c is much smaller than n (data size). More importantly, the user can control c by configuring the density threshold.

With an ordering of dense cells derived above, a very useful and interactive visualization can be derived. Let us look at an illustrative dataset shown in Figure 6 (top half). Imagine that each point there is a hyper-cell in a multidimensional space. The ordering of those cells is visualized in Figure 6 (bottom half). The horizontal axis represents the ordering of cells. The vertical axis represents the length of hyper-MST edges. Each vertical line segment is an edge in the hyper-MST of these cells and its height is the length of the edge. There is an edge between each pair of neighboring cells in the ordering. Note: as shown in Figure 3, the edge may not directly connect the two neighboring cells. Strictly speaking, each edge is connecting two clusters (i.e., two chains of cells).

With the visualization of cluster ordering depicted in Figure 6 and implemented as in Figure 7, a cluster appears as a valley in the graph. Distinct clusters are separated by long edges (high ridges). The dashed horizontal line (Figure 6) is the threshold value for cutting off long edges. By interactively dragging this threshold bar, one can interactively explore clusters at different hierarchical level. Clusters are automatically extracted and colored while the user moving the threshold bar. This visualization of cluster ordering can scale well with very large data sets, which is an advantage over traditional dendrograms. Using a real dataset, Figure 7 shows: (1) a density plot that supports interactive configuration of the density threshold, (2) a cluster ordering graph, (3) a trend plot (below the cluster ordering) that plots the relationship between the threshold value and the number of clusters identified (given a minimum cluster size), and (4) an HD cluster viewer based on a parallel coordinates plot. See Table 1

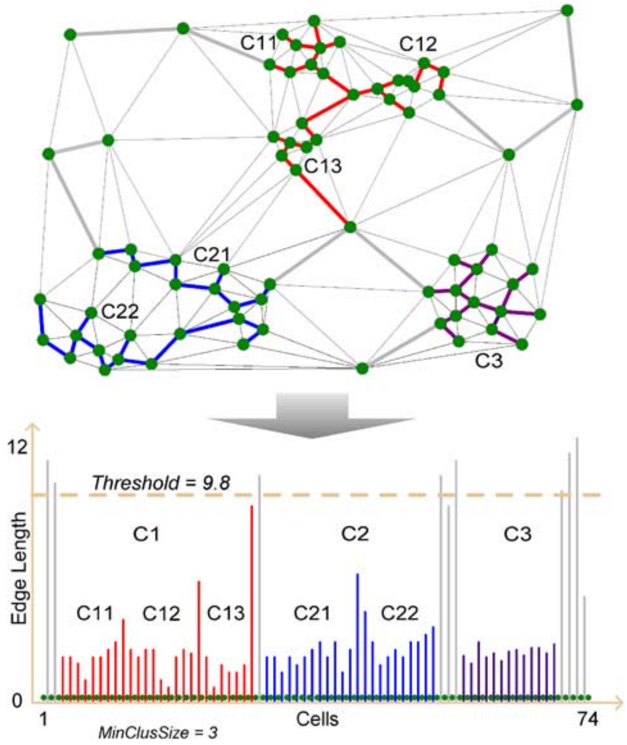


Figure 6 Deriving a cluster ordering from a graph of dense cells (here they are in a 2-D space for the convenience of demonstration).

and the penultimate section for further explanation of the data used here.

This visualization of cluster ordering is inspired by OPTICS[25] but has several significant differences from OPTICS. First, the cluster ordering in OPTICS is derived using a density- and neighborhood-based approach while the cluster ordering presented here is derived from a graph (although it is a graph of dense cells) and thus is applicable to visualize many existing hierarchical clustering methods (e.g., average- and complete-linkage methods). Second, each vertical line segment here represents a hyper-MST edge (not a point and its reachability as in OPTICS). Third, while OPTICS needs to identify potential start-of-cluster and end-of-cluster regions and then combine matching regions into clusters, the cluster ordering developed here makes it much easier for the user to perceive the cluster hierarchy and for the algorithm to automatically extract clusters.

In the literature there are also several other aggregation (or condensation) methods to summarize or group data points for efficient clustering analysis.[23,24,34] The incorporation and comparison of those condensation-based (or density-based) methods with the method introduced above is beyond the scope of this paper. The focus here is to introduce an overall framework and a suite of coordinated computational and visual methods, rather than evaluate a single clustering method.

Selection, coloring, linking, and brushing

Two groups of selection are supported: cell-based and cluster-based. A cell-based selection allows the user to select a single cell or a set of cells. In the HD cluster ordering and with the ‘Cell’ checkbox checked, the user can mouse-over any cell (the region between two neighboring lines) to highlight it, or drag the mouse to select a group of cells to highlight them. In the HD cluster viewer, with the ‘Indication’ box checked, the user can mouse-over a string to highlight it or drag the mouse to run across one or multiple strings to highlight. Cell-based selection can be further classified into two groups: union selection or intersect selection. With the ‘shift’ key down, the user can make multiple selections and highlight them all. It can also be called ‘addition’ selection. Similarly, in the HD cluster viewer. Intersect selection is supported only in the HD cluster viewer. The user can make the first selection as described above. Then check the ‘Intersect Sel.’ checkbox. Now the user can make another selection. Only those cells that are in both selections are highlighted.

A cluster-based selection allows the user to select all the cells of in a cluster. In the HD cluster viewer, clusters often partly overlap with each other and make it difficult to fully understand them. In the HD cluster ordering component, with the ‘Cluster’ checkbox checked, the user can mouse-over any cluster (cells between two neighboring lines that are longer or higher than the current threshold) to highlight all cells within that cluster. The user can also drag the mouse to select a group of cells that fall in the same cluster. This is possible in the HD cluster ordering because all cells in the same cluster are ordered contiguously.

Clusters are colored with a continuous color spectrum, which the user can interactively configure. The basic idea is that: the closer two clusters are, the more similar their colors should be. Since the HD cluster ordering is derived precisely to meet this requirement, colors are assigned to clusters according to their positions in the cluster ordering. Both selection operations and coloring of clusters are propagated among those visualization components, for example, HD cluster viewer, HD cluster ordering, and the map. Such linking and brushing can help the user to explore patterns from different but interrelated perspectives and achieve good understanding of both the data and discovered patterns (see Figure 9).

Component-Oriented Implementation

The implementation of the data exploration environment designed in this research adopts a component-oriented framework.[35] Each analysis approach that focuses on a specific analysis task, for example, feature selection, is implemented as an independent component – a JAVA Bean. Components that comply JAVA Bean specification can be easily integrated together within GeoVISTA *studio*, a JAVA-based visual programming environment.[36–39]

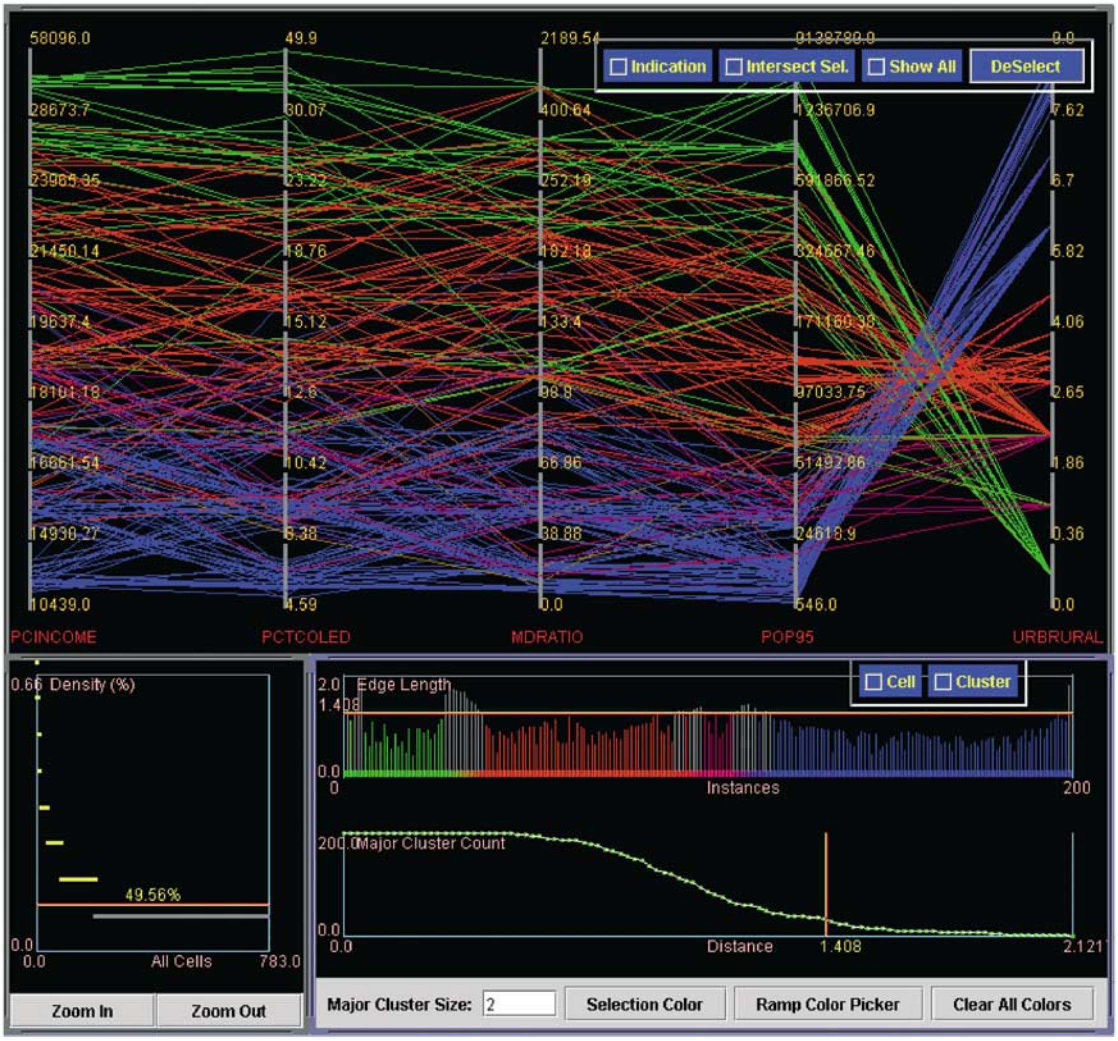


Figure 7 A density plot (bottom-left), a cluster ordering (bottom-right), a trend plot (below the cluster ordering), and an HD cluster viewer (top). See Table 1 and the section penultimate for further explanation of the data and patterns shown here.

The communication between components is based on the input/output interface (public methods within a JAVA bean) and events. An event is fired by a component when the component has changed something and wants other components to know about it. Other components can be notified of the event if they register as listeners to the component that fires the event. Different subsets of components can be assembled into a discovery system according to the need of a specific analysis task. Such a component-oriented implementation enables an open framework that is easy to add in new components or modify existing components to enhance the capability of the discovery environment.

Figure 8 shows a design in GeoVISTA *Studio* that assembles a suite of components into a knowledge discovery environment. Each component is represented using a rectangular icon, which has one or more inputs (red arrows) and produce one or more outputs (blue arrows). The user has the flexibility to select components and configure how they communicate with each other. In the design (Figure 8), four measures (correlation, χ^2, conditional entropy, and Kullback–Leibler divergence[40] are registered with the feature selection component. In this paper, only the conditional entropy and χ^2 measures are used. Two sorting methods, MST-based and null sorting (i.e., using original ordering of dimensions), are registered with the feature selection component for sorting variables with one of the above measures. Two interactive histograms are linked to the feature selection component for interactively classifying and coloring measure values in the entropy matrix. A DataCenter component (for loading data) is first linked to a data processing component (for preliminary examination and cleaning of the input data and variables), which is then linked to the feature selection component. A maximum subspace searcher component, an HD density plot component, an HD cluster ordering component, an HD cluster viewer, and a

Table 1 Attributes in the breast cancer dataset

Attribute	*Explanation*
Centroid.X	Coordinate X of the centroid of a county
Centroid.Y	Coordinate Y of the centroid of a county
BRRALLZ	Breast cancer mortality rate per 100,000 person-years, all races, all genders, all ages, for the time period 1970–1994
MDRATIO	# physicians per 100,000 population
HOSP	# hospitals per 100,000 population
PCTHISP	% of Hispanic origin
URBRURAL	USDA urban/rural code (0=most urban, 9=most rural)
PCINCOME	per capita income
PCTPOOR	% living below federal poverty line
PCTCOLED	% adults over 25 with 4+ years of college education
UNEMPLOY	% unemployed
POP95	1995 population
PAP3YRSM	% women aged 50–64 who had a pap test in past 3 years
MAMMOG2YSM	% women aged 50–64 who had a mammogram in past 2 years
OBESE	% of persons aged 18+ who are >120% of the median body mass index
NOINS	% of persons aged 18+ who do not have a health plan or health

map component are sequentially connected with the feature selection component.

Figure 9 shows a snapshot of the integrated system. A normal cycle within the iterative exploration process can be: loading data, cleaning the data, visualizing the matrix, using histograms to adjust colors for a better view of the matrix, interactively or automatically identify an interesting subspace, cutting the subspace into cells and selecting dense cells, deriving the cluster ordering of these dense cells, interactively exploring hierarchical clusters, visualizing the clusters in a map if spatial locations available.

Analyzing cancer data

The dataset used here contains 1156 counties with BRRALLZ (breast cancer mortality rate per 100,000 population for the time period 1970–1994) over 45. The dataset has 14 dimensions (Table 1) and two spatial dimensions – the centroid of each county. As introduced previously, the developed system can handle much more dimensions and larger data size than this application dataset. This dataset is under development in a funded research and more variables (e.g., factor variables from census data, environmental monitoring data, etc.) will be

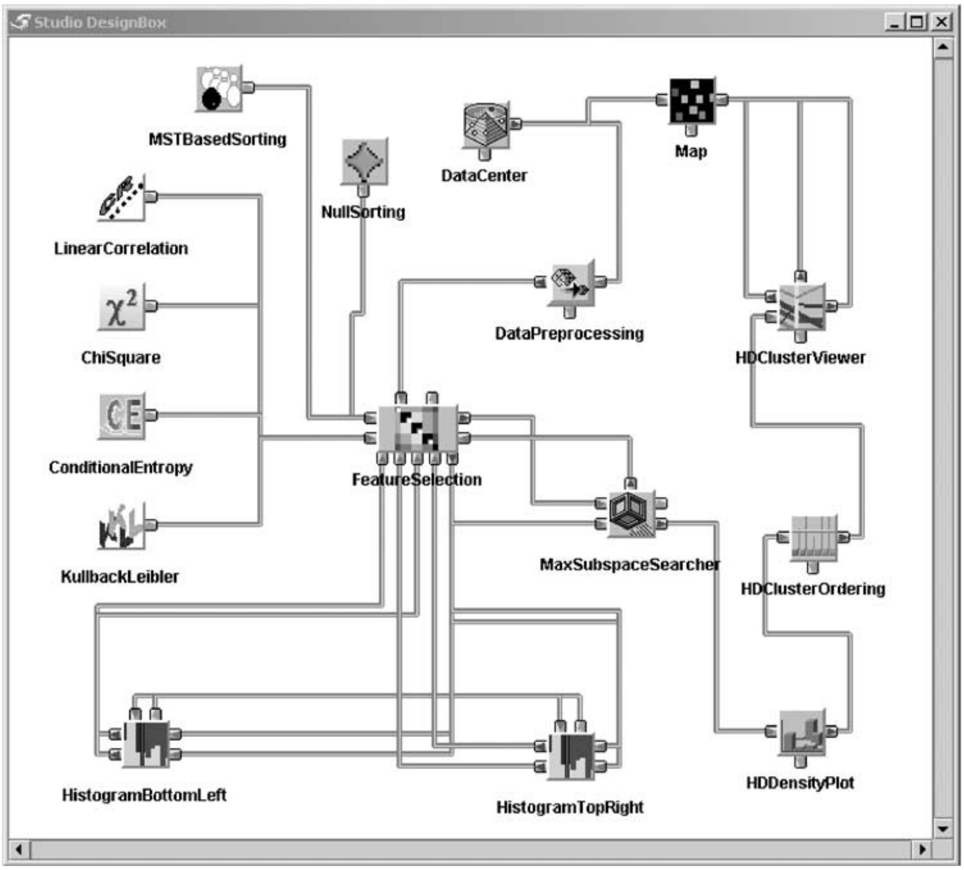
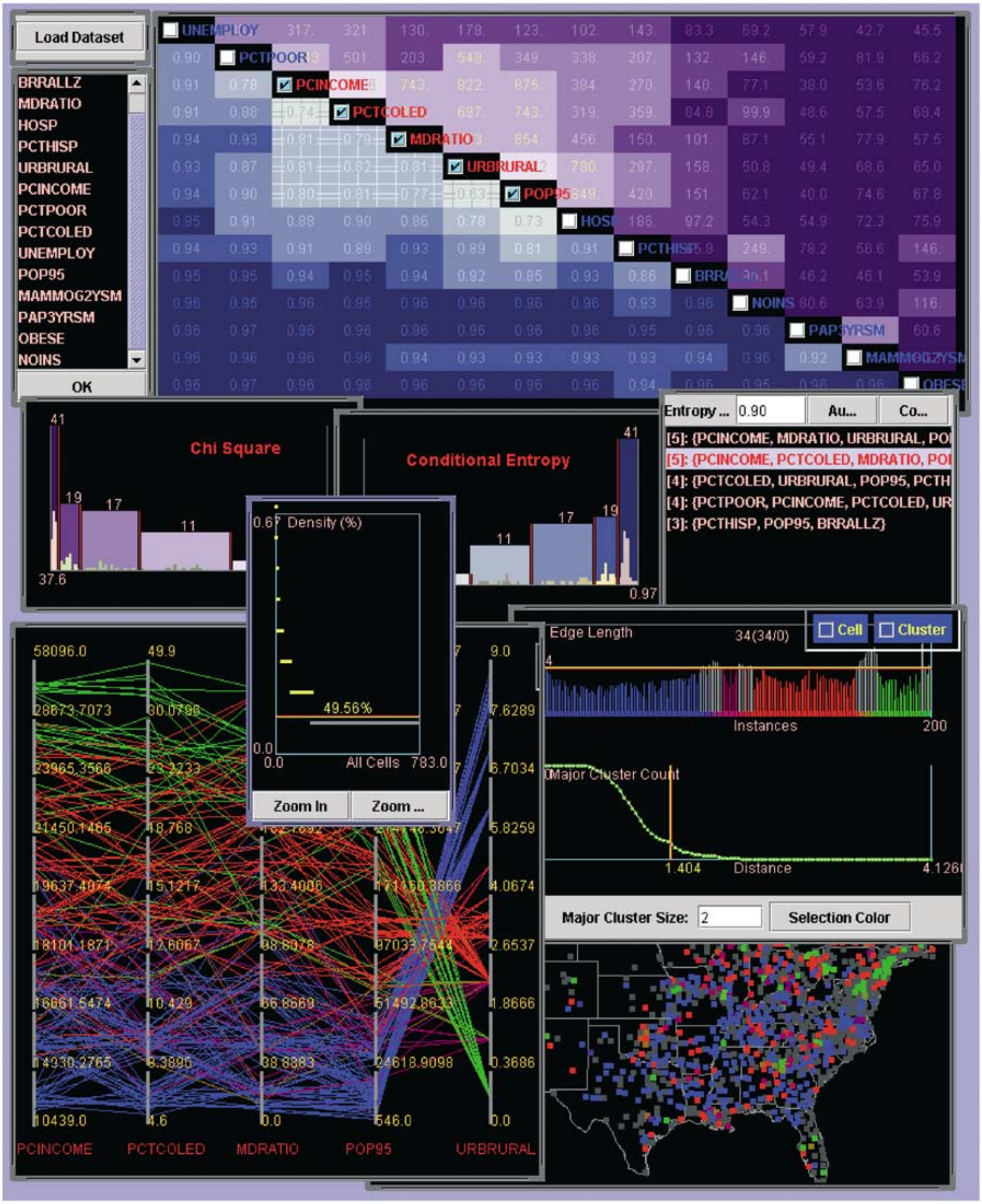


Figure 8 Assembling various components into a unified, collaborative working system. The wires between components specify how they communicate with each other.



Entropy ... 0.90

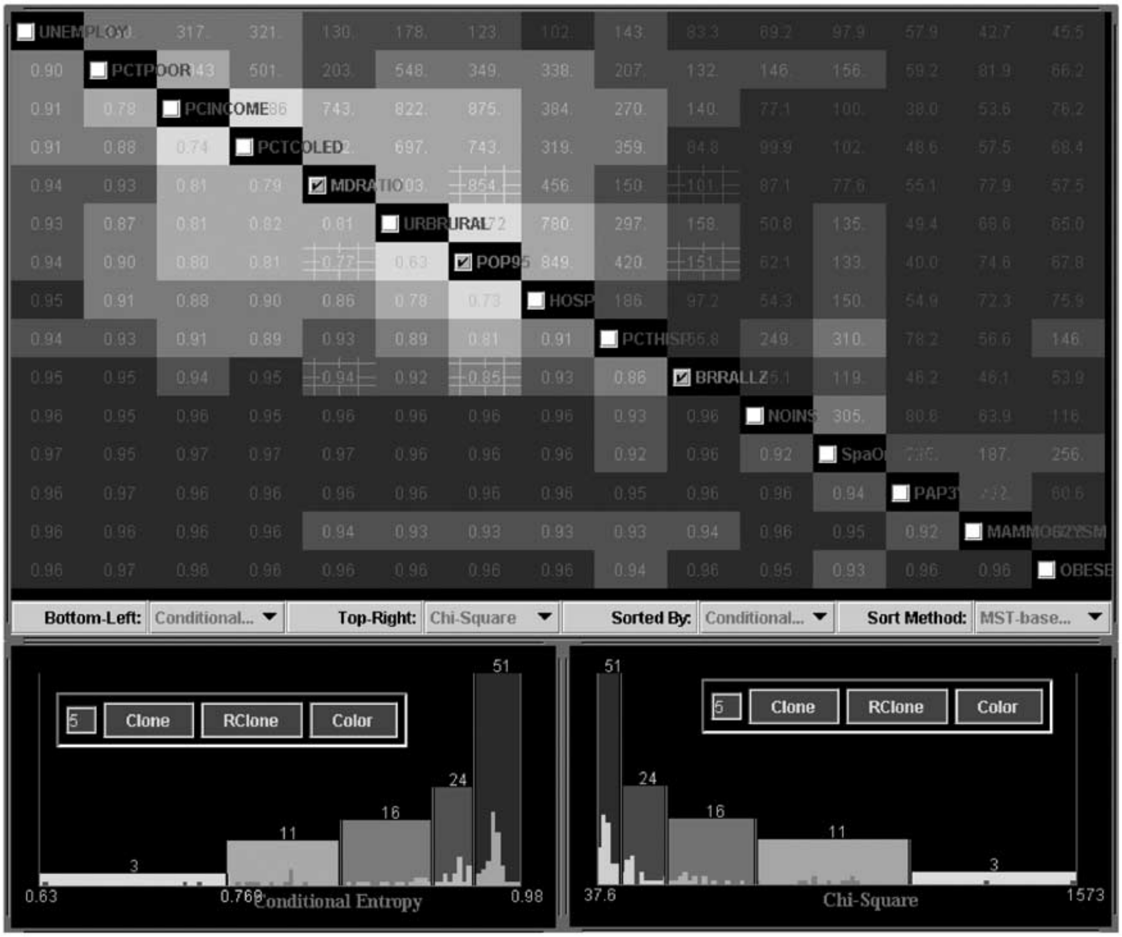
[5]: {PCINCOME, MDRATIO, URBRURAL, POI
[5]: {PCINCOME, PCTCOLED, MDRATIO, POI
[4]: {PCTCOLED, URBRURAL, POP95, PCTH
[4]: {PCTPOOR, PCINCOME, PCTCOLED, UR
[3]: {PCTHISP, POP95, BRRALLZ}

Major Cluster Size: 2

Figure 9 The interface of the integrated data exploration system.

added to study the potential causes and patterns for breast cancer incidences.

The conditional entropy matrix of all dimensions in the breast cancer data is shown in Figures 9 and 10. Either visually or automatically, {PCINCOME, PCTCOLED, MDRATIO, URBRURAL, POP95} appears to be the top ranked subspace that may have strong relationships. Once this subspace is selected (see Figure 9), it is passed to the HD density plot component, where the subspace is cut into 783 non-empty cells. Then a density threshold is interactively set and 200 dense cells are extracted. These 200 dense cells altogether contain about 50% of all data points (i.e., about 578 counties) (see Figures 7 and 9). These 200 dense cells are then passed to the HD cluster



☐ UNEMPLOY		317.	321.	130.	178.	123.	102.	143.	83.3	89.2	97.9	57.9	42.7	45.5
0.90	☐ PCTPOOR	43	501.	203.	548.	349.	338.	207.	132.	146.	156.	59.2	81.9	66.2
0.91	0.78	☐ PCINCOME	86	743.	822.	875.	384.	270.	140.	77.1	100.	38.0	53.6	76.2
0.91	0.88	0.74	☐ PCTCOLED	2.	697.	743.	319.	359.	84.8	99.9	102.	48.6	57.5	68.4
0.94	0.93	0.81	0.79	☑ MDRATIO	03.	854.	456.	150.	101.	87.1	77.6	55.1	77.9	57.5
0.93	0.87	0.81	0.82	0.81	☐ URBRURAL	72	780.	297.	158.	50.8	135.	49.4	68.6	65.0
0.94	0.90	0.80	0.81	0.77	0.63	☑ POP95	849.	420.	151.	62.1	133.	40.0	74.6	67.8
0.95	0.91	0.88	0.90	0.86	0.78	0.73	☐ HOSP	186.	97.2	54.3	150.	54.9	72.3	75.9
0.94	0.93	0.91	0.89	0.93	0.89	0.81	0.91	☐ PCTHISP	65.8	249.	310.	78.2	56.6	146.
0.95	0.95	0.94	0.95	0.94	0.92	0.85	0.93	0.86	☑ BRRALLZ	5.1	119.	46.2	46.1	53.9
0.96	0.95	0.96	0.95	0.96	0.96	0.96	0.96	0.93	0.96	☐ NOINS	305.	80.6	63.9	116.
0.97	0.95	0.97	0.97	0.97	0.96	0.96	0.96	0.92	0.96	0.92	☐ SpaO	[illegible]	187.	256.
0.96	0.97	0.96	0.96	0.96	0.96	0.96	0.96	0.95	0.96	0.96	0.94	☐ PAP3	32.	60.6
0.96	0.96	0.96	0.96	0.94	0.93	0.93	0.93	0.93	0.94	0.96	0.95	0.92	☐ MAMMOG2YSM	
0.96	0.97	0.96	0.96	0.96	0.96	0.96	0.96	0.94	0.96	0.95	0.93	0.96	0.96	☐ OBESE

Bottom-Left: Conditional... Top-Right: Chi-Square Sorted By: Conditional... Sort Method: MST-base...

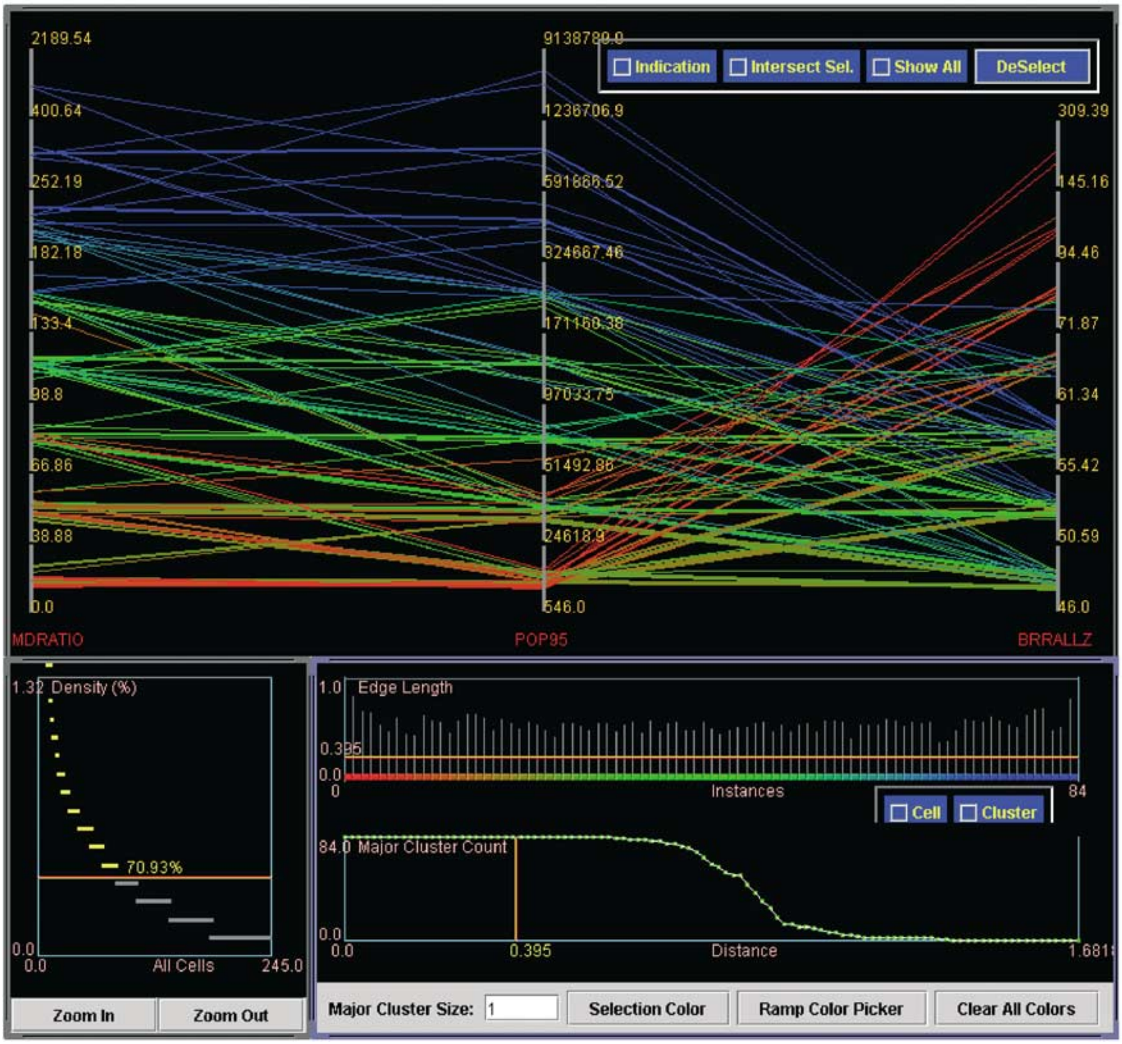
Figure 10 Visualizing the entropy matrix of a cancer dataset (with χ^2 values above the diagonal). Subspace {BRRALLZ, POP95, MDRATIO} is selected.

ordering component, where an ordering and visualization of these dense cells are derived (see Figures 7 and 9). The user can then interactively change the threshold to see clusters at different levels. The change of clusters is immediately propagated to the HD cluster viewer and the map (see Figure 9). In the snapshot shown in Figure 7, there are four major clusters: green, red, purple and blue. The strong relationships among these five variables can be easily seen and interpreted. Their spatial distribution is presented in the map.

However, these patterns may seem obvious and trivial to some users. The relationship between POP95 and URBRURAL is also obvious because the URBRURAL value is pretty much derived using POP95. Indeed, due to the complex and various relationships in a real dataset, human expertise and interpretation are indispensable in guiding the whole discovery process. For the knowledge discovery environment, being able to find the obvious first is actually a good thing because the obvious is often the strongest patterns in the data. What is needed is to either use human interaction or encode human knowledge in the system to 'ignore' the obvious and focus on unexpected, novel patterns.

If the user is interested in studying various relationships between the breast cancer mortality rate and other variables, she/he may force the BRRALLZ variable to be included in the selected subspace. From the matrix, we can see that BRRALLZ is strongly associated with POP95, URBRURAL, and MDRATIO. Since we already know the strong (but obvious) linear relationship between POP95 and URBRURAL, only one of them (here it is POP95) is included in the selected subspace {BRRALLZ, POP95, MDRATIO} (see Figure 10). The subspace {BRRALLZ, POP95, MDRATIO} is cut into 245 non-empty cells. With the density threshold set in Figure 11, 84 dense cells are extracted, which contain 70.93% of all data pints (820 counties).

These dense cells are then passed to the HD cluster ordering component. Although the clustering ordering of these cells does not show a clear hierarchical cluster structure, the color assigned to each cell based on the ordering still uncovers strong patterns in the HD cluster viewer and the map (see Figures 11 and 12). From colors in the HD cluster viewer and the map, we can see mainly four clusters: blue, green, red, and light brown. The blue cluster represents those counties with high population,



Major Cluster Size: 1

Figure 11 The density plot, HD cluster ordering, and the HD cluster viewer with subspace {BRRALLZ, POP95, MDRATIO}.

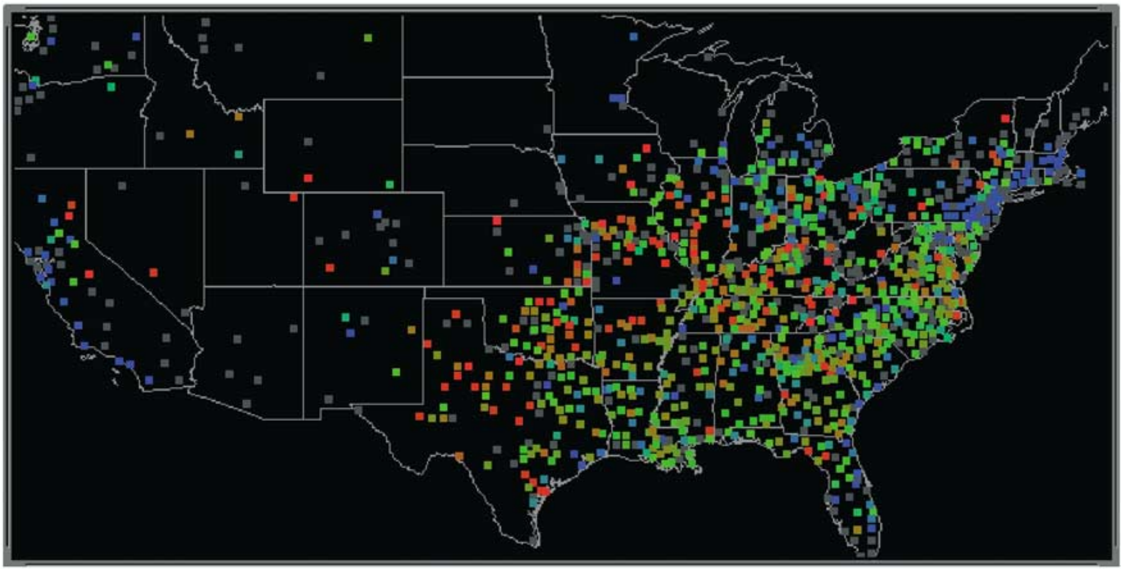


Figure 12 Mapping the patterns found in subspace {BRRALLZ, POP95, MDRATIO}.

high doctor ratio, and slightly-below-average breast cancer ratio. The green cluster represents those counties with average doctor ratio, average-low population, and low breast cancer ratio. The red cluster represent those counties with very low doctor ratio, very low population and high breast cancer ratio! The light brown cluster is between the green and the red, representing counties that have low doctor ratio, low population, and low-average breast cancer ratio.

The map of these clusters also shows interesting spatial patterns (Figure 12). One can see that counties within the red cluster also spatially clustered in mid-states. Another exciting feature of the coloring is that it enables us to see the transition between clusters, both in the multidimensional space shown with the HD cluster viewer and the geographic space shown with the map. Although for this case the subspace has only three variables (and thus a traditional 3-D scatterplot can be employed to visualize patterns in it), the developed system enables a unified approach for searching patterns in subspaces of various dimensionalities (e.g., in Figures 7 and 9, the subspace under examination is 5-D).

Conclusion

The goal of this research is to develop a highly interactive analysis environment that integrates the best of both human and machine capabilities for exploring large and high-dimensional data. Specifically, the research includes: (1) an interactive feature selection method for identifying interesting subspaces, (2) an interactive, hierarchical clustering method for searching arbitrary-shaped multidimensional clusters, and (3) a suite of coordinated visualization and computational components centered around the above two methods to facilitate an efficient and effective human-led exploration of high-dimensional data. The overall analysis is by nature an iterative process.

The research shows that computational approaches and visualization tools not only can be used together in a tightly coupled manner, but that used in this manner, each can also enhance and improve the capabilities of the other. For example, the cluster ordering greatly improves the visual presentation and coloring in visualization components. The aggregation of data points into cells makes visualization components more efficient and clearer for presenting and exploring patterns. Various visualization components, on the other hand, can help on configuring computation algorithms, interpreting patterns, and guiding the overall process of exploration. The application in analyzing the cancer dataset shows that the developed system can efficiently and effectively assist people in exploring high-dimensional data and identifying unknown (even unexpected) patterns.

Acknowledgments

Research presented in this paper was partially funded by the National Science Foundation (NSF) grant #9983445, NSF grant #EIA-9983451, and grant CA95949 from the National Cancer Institute (NCI). The author is grateful to Dr. Donna J. Peuquet, Dr. Mark Gahegan, and Dr. Alan M. MacEachren for their inputs and support in developing this research.

References

1 Agrawal R, Gehrke J, Gunopulos D, Raghavan P. *Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications*. ACM SIGMOD International Conference on Management of Data (Seattle, WA, U.S.A., 1998), ACM Press: New York, 94–105.

2 Procopiuc CM, Jones M, Agarwal PK, Murali TM. *A Monte Carlo Algorithm for Fast Projective Clustering*. ACM SIGMOD International Conference on Management of Data (Madison, WI, U.S.A., 2002), ACM Press: New York, 418–427.

3 Cheng C, Fu A and Zhang Y. *Entropy-based subspace clustering for mining numerical data*. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (San Diego, CA, USA, 1999), ACM Press: New York, 84–93.

4 Fayyad U, Piatetsky-Shapiro G, Smyth P. *From data mining to knowledge discovery-an review*. In: Fayyad U, Piatetsky-Shapiro G, Smyth P, Uthurusay R (Eds). Advances in Knowledge Discovery. AAAI Press/The MIT Press: Cambridge, MA, 1996; 1–33.

5 Liu H, Motoda H. *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic Publishers: Dordrecht, 1998; 214pp.

6 Dy JG, Brodley CE. *Feature subset selection and order identification for unsuprervised learning*. The Seventeenth International Conference on Machine Learning, Stanford University (CA, U.S.A., 2000), 247–254.

7 Dy JG, Brodley CE. *Visualization and interactive feature selection for unsupervised data*. The Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (Boston, MA, U.S.A., 2000), ACM Press: New York, 360–364.

8 Kim Y, Street WN, Menczer F. *Feature selection in unsupervised learning via evolutionary search*. The Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (Boston, MA, U.S.A., 2000), ACM Press: New York, 365–369.

9 Aggarwal CC, Wolf JL, Yu PS, Procopiuc C, Park JS. *Fast algorithms for projected clustering*. ACM SIGMOD International Conference on Management of Data (Philadelphia, Pennsylvania, U.S.A., 1999), ACM Press: New York, 61–72.

10 Aggarwal CC, Yu PS. *Finding generalized projected clusters in high dimensional spaces*. ACM SIGMOD International Conference on Management of Data (Dallas, TX, U.S.A, 2000), ACM Press: New York, 70–81.

11 Jong Hd, Rip A. *The computer revolution in science: steps toward the realization of computer-supported discovery environments*. Artificial Intelligence 1997; **91**: 225–256.

12 Wong PC. *Visual data mining*. IEEE Computer Graphics & Applications 1999; **19**: 20–31.

13 Ankerst M, Ester M, Kriegel H-P. *Towards an effective cooperation of the user and the computer for classification*. The Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (Boston, MA, USA, 2000), ACM Press: New York, 179–188.

14 Kohonen T. *Self-Organizing Maps*. Springer: Berlin, 2001, 501pp.

15 Friendly M. *Corrgrams: exploratory displays for correlation matrices*. The American Statistician 2002; **19**: 316–325.

16 Ankerst M, Berchtold S, Keim DA. *Similarity clustering of dimensions for an enhanced visualization of multidimensional data*. Information Visualization '98 (Raleigh-Durham, NC, USA, 1998), 52–60.

17 Gordon AD. *A review of hierarchical classification*. Journal of the Royal Statistical Society. Series A (General) 1987; **150**: 119–137.

18 Gordon AD. *Hierarchical classification.* In: Arabie P, Hubert LJ, Soete GD (Eds). Clustering and Classification. World Scientific Publisher: River Edge, NJ, USA; 1996; 65–122.

19 Jain AK, Dubes RC. *Algorithms for Clustering Data*. Prentice-Hall: Englewood Cliffs, NJ, 1988, 320pp.

20 Jain AK, Murty MN, Flynn PJ. *Data clustering: a review*. ACM Computing Surveys (CSUR) 1999; **31**: 264–323.

21 Guo D, Peuquet D, Gahegan M. *ICEAGE: interactive clustering and exploration of large and high-dimensional geodata*. GeoInformatica 2003; **7**: 229–253.

22 Ester M, Kriegel H-P, Sander J, Xu X. *A density-based algorithm for discovering clusters in large spatial databases with noise*. The Second International Conference on Knowledge Discovery and Data Mining (Portland, OR, USA, 1996), AAAI Press: New York, 226–231.

23 Schikuta E. *Grid-clustering: An efficient hierarchical clustering method for very large data sets*. 13th Conference on Pattern Recognition, 1996, IEEE Computer Society Press: New York, 101–105.

24 Hinneburg A, Keim DA. *Optimal grid-clustering: towards breaking the curse of dimensionality in high-dimensional clustering*. The 25th VLDB Conference (Edingburgh, Scotland, 1999), Morgan Kaufmann: Los Altos, CA, 506–517.

25 Ankerst M, Breunig MM, Kriegel H-P, Sander J. *OPTICS: Ordering Points To Identify the Clustering Structure*. ACM SIGMOD International Conference on Management of Data (Philadelphia, PA, USA, 1999), ACM Press: New York, 49–60.

26 Hinneburg A, Keim DA, Wawryniuk M. *HD-Eye: Visual mining of high-dimensional data*. IEEE Computer Graphics & Applications 1999; **19**: 22–31.

27 Seo J, Shneiderman B. *Interactively exploring hierarchical clustering results [gene identification]*. Computer 2002; **35**: 80–86.

28 Guo D, Gahegan M, Peuquet D, MacEachren A. *Breaking down dimensionality: an effective feature selection method for high-dimensional clustering*. Workshop on Clustering High Dimensional Data and its Applications, the Third SIAM International Conference on Data Mining, May 1 – 3 (San Francisco, CA, U.S.A., 2003).

29 Pyle D. *Data preparation for data mining*. Morgan Kaufmann: Los Altos, CA, 1999, 540pp.

30 Snedecor GW, Cochran WG. *Statistical Methods*. Iowa State University Press: IA, U.S.A. 1989, 503pp.

31 Vandev D, Tsvetanova GY. *Perfect chains and single linkage clustering algorithm*. Statistical Data Analysis, Proceedings SDA-95, 1995; 99–107.

32 Duda RO, Hart PE, Stork DG. *Pattern Classification*. John Wiley & Sons: New York, 2001.

33 Guo D, Peuquet D, Gahegan M. *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, 2002), 131–136.

34 Zhang T, Ramakrishnan R, Livny M. *BIRCH: an efficient data clustering method for very large databases*. ACM SIGMOD International Conference on Management of Data (Montreal, Canada, 1996), ACM Press: New York, 103–114.

35 Szyperski C. *Component Software: Beyond Object-Oriented Programming*. Addison-Wesley/ACM Press: New York, 1999, 589pp.

36 Gahegan M, Takatsuka M, Wheeler M, Hardisty F. *GeoVISTA Studio: a geocomputational workbench*. 5th International Conference on GeoComputation, University of Greenwich (Medway Campus, U.K., 2000).

37 Gahegan M, Takatsuka M, Wheeler M, Hardisty F. *Introducing GeoVISTA Studio: an integrated suite of visualization and computational methods for exploration and knowledge construction in geography*. Computers, Environment and Urban Systems 2001; **26**: 267–292.

38 MacEachren AM, Hardisty F, Gahegan M, Wheeler M, Dai X, Guo D, Takatsuka M. *Supporting visual integration and analysis of geospatially-referenced statistics through web-deployable, cross-platform tools*. dg.o.2001, National Conference for Digital Government Research (Los Angeles, CA, 2001), 17–24.

39 MacEachren AM, Hardisty F, Dai X, Pickle L. *Supporting visual analysis of federal statistical summaries*. Communications of the ACM 2003; **46**: 59–60.

40 Kullback S, Leibler RA. *On information and sufficiency*. The Annals of Mathematical Statistics 1951; **22**: 79–86.