Information Visualization (2005) 4, 96–113

www.palgrave-journals.com/ivs

A rank-by-feature framework for interactive exploration of multidimensional data

Jinwook Seo[1,2]
Ben Shneiderman[1,2,3]

[1]*Department of Computer Science, University of Maryland, College Park, MD 20742, U.S.A.;*
[2]*Human–Computer Interaction Lab, Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742, U.S.A.;*
[3]*Institute for Systems Research, University of Maryland, College Park, MD 20742, U.S.A.*

Correspondence: Ben Shneiderman, Department of Computer Science, A.V. Williams Building, College Park, MD 20742, U.S.A.
Tel: +1 301-405-2680,
Fax: +1 301-405-6707
E-mail: ben@cs.umd.edu

Received: 30 October 2004
Revised: 15 January 2005
Accepted: 1 February 2005
Online publication date: 19 May 2005

Abstract

Interactive exploration of multidimensional data sets is challenging because: (1) it is difficult to comprehend patterns in more than three dimensions, and (2) current systems often are a patchwork of graphical and statistical methods leaving many researchers uncertain about how to explore their data in an orderly manner. We offer a set of principles and a novel rank-by-feature framework that could enable users to better understand distributions in one (1D) or two dimensions (2D), and then discover relationships, clusters, gaps, outliers, and other features. Users of our framework can view graphical presentations (histograms, boxplots, and scatterplots), and then choose a feature detection criterion to rank 1D or 2D axis-parallel projections. By combining information visualization techniques (overview, coordination, and dynamic query) with summaries and statistical methods users can systematically examine the most important 1D and 2D axis-parallel projections. We summarize our Graphics, Ranking, and Interaction for Discovery (GRID) principles as: (1) study 1D, study 2D, then find features (2) ranking guides insight, statistics confirm. We implemented the rank-by-feature framework in the Hierarchical Clustering Explorer, but the same data exploration principles could enable users to organize their discovery process so as to produce more thorough analyses and extract deeper insights in any multidimensional data application, such as spreadsheets, statistical packages, or information visualization tools.
Information Visualization (2005) **4,** 96–113. doi:10.1057/palgrave.ivs.9500091

Keywords: Rank-by-feature framework; information visualization; exploratory data analysis; dynamic query; feature detection/selection; graphical displays

Introduction

Multidimensional or multivariate data sets are common in data analysis applications; e. g., microarray gene expression, demographics, and economics. A data set that can be represented in a spreadsheet where there are more than three columns can be thought of as multidimensional. Without losing generality, we can assume that each column is a dimension (or a variable), and each row is a data item. Dealing with multidimensionality has been challenging to researchers in many disciplines due to the difficulty in comprehending more than three dimensions to discover relationships, outliers, clusters, and gaps. This difficulty of navigating in the sparseness of high dimensional spaces is so well recognized that it has a provocative name: 'the curse of high dimensionality.'

One of the commonly used methods to cope with multidimensionality is to use low-dimensional projections. Since human eyes and minds are effective in understanding one-dimensional (1D) histograms, two-dimensional (2D) scatterplots, and three-dimensional (3D) scatterplots, these representations are often used as a starting point. Users can begin by

understanding the meaning of each dimension (since names can help dramatically, they should be readily accessible) and by examining the range and distribution (normal, uniform, erratic, etc.) of values in a histogram. Then experienced analysts suggest applying an orderly process to note exceptional features such as outliers, gaps, or clusters.

Next, users can explore 2D relationships by studying 2D scatterplots and again use an orderly process to note exceptional features. Since computer displays are intrinsically two-dimensional, collections of 2D projections have been widely used as representations of the original multidimensional data. This is imperfect since some features will be hidden, but at least users can understand what they are seeing and come away with some insights.

Advocates of 3D scatterplots argue that since the natural world is three dimensional, users can readily grasp 3D representations. However, there is substantial empirical evidence that for multidimensional ordinal data (rather than 3D real objects such as chairs or skeletons), users struggle with occlusion and the cognitive burden of navigation as they try to find desired viewpoints. Advocates of higher dimensional displays have demonstrated attractive possibilities, but their strategies are still difficult to grasp for most users.

Since 2D presentations offer ample power while maintaining comprehensibility, many variations have been proposed. We distinguish the three categories of 2D presentations by the way axes are composed: (1) Non-axis-parallel projection methods use a (linear/nonlinear) combination of two or more dimensions for an axis of the projection plane. Principal component analysis (PCA) is a well-established technique in this category, (2) Axis-parallel projection methods often called marginal views use existing dimensions as axes of the projection plane. One of the existing dimensions is selected as the horizontal axis, and another as the vertical axis, to make a familiar and comprehensible presentation. Sometimes, other dimensions can be mapped as color, size, length, angle, etc., (3) Novel methods use axes that are not directly derived from any combination of dimensions. For example, the parallel coordinate presentation is a powerful concept in which dimensions are aligned sequentially and presented perpendicular to a horizontal axis.[1] Self organizing maps (SOM)[4] are also in this category.

Although presentations in category (1), non-axis-parallel, can show all possible 2D projections of a multidimensional data set, they suffer from users' difficulty in interpreting 2D projections whose axes are linear/nonlinear combination of two or more dimensions. For example, even though users may find a strong linear correlation on a projection where the horizontal axis is 3.7*body weight* - 2.3*height* and the vertical axis is *waist size* $+ 2.6$*chest size*, the finding is not so useful because it is difficult to understand the meaning of such projections.

Techniques in category (2), axis-parallel, have a limitation that features can be detected in only the two selected dimensions. However, since it is familiar and comprehensible for users to interpret the meaning of the projection, these projections have been widely used and implemented in visualization tools. A problem with these category (2) presentations is how to deal with the large number of possible low-dimensional projections. If we have an m-dimensional data set, we can generate $m(m-1)/2$ 2D projections using the category (2) techniques. We believe that our work offers an attractive solution to coping with the large numbers of low-dimensional projections and that it provides practical assistance in finding features in multidimensional data.

Techniques in category (3) remain important, because many relationships and features are visible and meaningful only in higher dimensional presentations. Our principles could be applied to support these techniques as well, but that subject is beyond this paper's scope.

There have been many commercial packages and research projects that utilize low-dimensional projections for exploratory data analysis, including spreadsheets, statistical packages, and information visualization tools. However, users have to develop their own strategies to discover which projections are interesting and to display them. We believe that existing packages and projects, especially information visualization tools for exploratory data analysis, can be improved by enabling users to systematically examine low-dimensional projections.

In this paper, we present a conceptual framework for interactive feature detection named *rank-by-feature framework* to address these issues. In the rank-by-feature framework (the rank-by-feature interface for 2D scatterplots is shown at the bottom half of Figure 1), users can select an interesting ranking criterion, and then all possible axis-parallel projections of a multidimensional data set are ranked by the selected ranking criterion. Available ranking criteria are explained in the subsection in ID Histogram ordering and 2D scatterplot ordering. The ranking result is visually presented in a color-coded grid ('Score Overview'), as well as a tabular display ('Ordered List') where each row represents a projection and is color-coded by the ranking score. With these presentations users can easily perceive the most interesting projections, and also grasp the overall ranking score distribution. Users can manually browse projections by rapidly changing the dimension for an axis using the item slider attached to the corresponding axis of the projection view (histogram and boxplot for 1D, and scatterplot for 2D).

For example, let's assume that users analyze the US counties data set with 17 demographical and economical statistics available for each county. The data set can be thought of as a 17 dimensional data set. Users can choose 'Pearson's correlation coefficient' as a ranking criterion at the rank-by-feature framework if they are interested in linear relationships between dimensions. Then, the rank-by-feature framework calculates 'scores' (in this case,

Figure 1 The Hierarchical Clustering Explorer (HCE) with a US counties statistics data set. The interactively coordinated displays in HCE 3.0 include: dendrogram view, histogram views, scatterplot views, details view at the top, and seven tabs (Color Mosaic, Table View, Histogram Ordering, Scatterplot Ordering, Profile Search, Gene Ontology, and K-means) at the bottom (Scatterplot Ordering tab is selected in this figure). The color mosaic (shown in red and green in the top left) has one horizontal row for every column of input data. Each county has 17 variables appearing as a vertical stripe of red and green cells. The dendrogram view at the top left corner visualizes the hierarchical clustering result of a US counties statistics data set enabling users to interactively explore the clustering result,[2] Among the seven tabs, Histogram Ordering and Scatterplot Ordering implement the rank-by-feature framework interface for 1D and 2D, respectively. Two histograms and two scatterplots are selected through the rank-by-feature interfaces and are shown as separate child windows to the right of the dendrogram view. Four selected US counties are listed in the top half of the details view and the statistics for one of the counties are shown at the bottom half. A 2D scatterplot ordering result using 'Pearson's correlation coefficient' as the ranking criterion is shown in the Scatterplot Ordering tab. Four counties that are poor and have a medium number of high-school graduates are selected in the scatterplot browser and they are all highlighted in other views with triangles.

Pearson's correlation coefficient) for all possible pair of dimensions, and ranks all pairs according to the score values. Users could easily identify that there is a negative correlation between poverty level and the percentage of high school graduates after users skim through the score overview (a color-coded grid display at the lower left corner of Figure 1), where each cell represents the scatterplot for a pair of dimensions and it is color-coded by the score value for the scatterplot. All possible pairs are also shown in the ordered list (a list control right next to the score overview at Figure 1) together with the numerical score values in a column. The scatterplot is shown at the lower right corner of Figure 1. More details on the rank-by-feature framework are explained in the subsequent section. More details on the application example of the rank-by-feature framework with the US counties data set are explained in the section in Application example.

We implemented the rank-by-feature framework in our interactive exploration tool for multidimensional data, the Hierarchical Clustering Explorer (HCE)[2] (Figure 1) as two new tab windows ('Histogram Ordering' for 1D projections, and 'Scatterplot Ordering' for 2D projections). By using the rank-by-feature framework, users can easily find interesting histograms and scatterplots, and generate separate windows to visualize those plots. All these plots are interactively coordinated with other views (e.g. dendrogram and color mosaic view, tabular view, parallel coordinate view) in HCE. If users select a group of items in any view, they can see the selected items highlighted in all other views. Thus, it is possible to comprehend the data from various perspectives to get more meaningful insights.

This paper is an extended version of our paper for the Information Visualization Conference, Austin Texas, 2004.[3] We extend the two basic statistical principles for exploratory data analysis to encompass the interactive visualizations and user interactions, and we present our principles for interactive multidimensional data exploration – Graphics, Ranking, and Interaction for Discovery (GRID) principles. We improve the color coding scheme for the rank-by-feature framework by using three different colors and we overhaul the visualization of features in 1D and 2D projections by highlighting key features appropriately. More ranking criteria are implemented in HCE and all ranking criteria are discussed in more detail in terms of why they are important and how to detect them. We also discuss the issues of transformation and other potential ranking criteria.

The next section introduces related work, and the subsequent section makes the case for the GRID principles and the rank-by-feature framework for axis-parallel 1D and 2D projections. Potentially interesting ranking criteria and transformations are discussed in the forthcoming section. Two application examples are presented in the next section. Discussion and future work are in the penultimate section. Finally, we conclude the paper in the last section.

Related work

Two-dimensional projections have been utilized in many visualization tools and graphical statistics tools for multidimensional data analysis. Projection techniques such as PCA and multidimensional scaling (MDS) are used to find informative 2D projections of multidimensional data sets. Examining only a single projection for a multidimensional data set is not enough to discover all the interesting features in the original data since any one projection may obscure some features.[5] Thus it is inevitable that users must scrutinize a series of projections to reveal the features of the data set.

Therefore statisticians developed strategies such as the grand tour[7] to systematically explore all possible linear projections of high dimensional data onto lower dimensions, most commonly 2D projections. In a grand tour users see a movie-like animation that has an equal possibility of showing all 2D projections. However, depending on the complexity of the distributions, it might take many minutes or several hours to complete a visual search in four dimensions.[8] As the number of dimensions grows, an exhaustive visual search is out of the question.

To alleviate this problem, various forms of guided tours, known as projection pursuit, attempt to lead viewers to the most interesting projections early in the tour. When projecting high dimension data onto 2D, the algorithm computes the projection pursuit index for all feasible 2D projections and leads viewers to the highest index projections. Animated transitions among these interesting projections show viewers the context and highlight the high index projections. Friedman and Tukey[5,6] believed that deviation from the normal distribution was a useful projection pursuit index. Their ideas were implemented in the pioneering visualization package XGobi.[9] This package was succeeded by GGobi,[31] which implements both the grand tour and the projection pursuit, but not the ranking that we propose.

There are clustering methods that utilize a series of low-dimensional projections in category (1). Among them, HD-Eye system by Hinneburg *et al.*[10] implements an interactive divisive hierarchical clustering algorithm built on a partitioning clustering algorithm, or OptiGrid. They show projections using glyphs, color or curve-based density displays to users so that users can visually determine low-dimensional projections where well-separated clusters are and then users can define separators on the projections.

These automatic projection pursuit methods made impressive gains in the problem of multidimensional data analysis, but they have limitations. One of the most important problems is the difficulty in interpreting the solutions from the automatic projection pursuit. Since the axes are the linear combination of the variables (or dimensions) of the original data, it is hard to determine what the projection actually means to users. Conversely, this is one of the reasons that axis-parallel projections

(projection methods in category (2)) are used in many multidimensional analysis tools.[11–13]

Projection methods belonging to category (2), axis-parallel, have been applied by researchers in machine learning, data mining, and information visualization. In machine learning and data mining, ample research has been conducted to address the problems of using projections. Most work focuses on the detection of dimensions that are most useful for a certain application, for example, supervised classification. In these areas, the term 'feature selection' is a process that chooses an optimal subset of features according to a certain criterion,[14] where a feature simply means dimension. Basically, the goal is to find a good subset of dimensions (or features) that contribute to the construction of a good classifier. Unsupervised feature selection methods are also studied in close relation with unsupervised clustering algorithms. In this case, the goal is to find an optimal subset of features with which clusters are well identified.[11,15–17] In pattern recognition, researchers want to find a subset of dimensions with which they can better detect specific patterns in a data set. In subspace-based clustering analysis, researchers want to find projections where it is easy to naturally partition the data set.

In the information visualization field, about 30 years ago, Jacques Bertin[18] presented a visualization method called the Permutation Matrix.[6] It is a reorderable matrix where a numerical value in each cell are represented as a graphical object whose size is proportional to the numerical value, and where users can rearrange rows and columns to get more homogeneous structure. This idea seems trivial, but it is a powerful way to observe meaningful patterns after rearranging the order of the data presentation. Since then, other researchers have also tried to optimally arrange dimensions so that similar or correlated dimensions are put close to each other. This helps users to find interesting patterns in multidimensional data.[19–21] Yang *et al*[21] proposed innovative dimension ordering methods to improve the effectiveness of visualization techniques including the parallel coordinates view in category (3). They rearrange dimensions within a single display according to similarities between dimensions or relative importance defined by users. Our work is to rank all dimensions or all pairs of dimensions whose visualization contains desired features. Since our work can provide a framework where statistical tools and algorithmic methods can be incorporated into the analysis process as ranking criteria, we think our work contributes to the advance of information visualization systems by bridging the analytic gaps that were recently discussed by Amar and Stasko.[22]

In early 1980s, Tukey who was one of the prominent statisticians who foresaw the utility of computers in exploratory data analysis envisioned a concept of 'scagnostics' (a special case of 'cognostics' – computer guiding diagnostics).[23] With high-dimensional data, it is necessary to use computers to evaluate the relative interest of different scatterplots, or the relative importance of showing them and sort out such scatterplots for human analyses. He emphasized the need for better ideas on 'what to compute' and 'how' as well as 'why.' He proposed several scagnostic indices such as the projection-pursuit clottedness and the difference between the classical correlation coefficient and a robust correlation. We brought his concept to reality with the rank-by-feature framework in the Hierarchical Clustering Explorer where we create interface controls, design practical displays, and implement more ranking ideas. There are also some research tools and commercial products for helping users to find more informative visualizations. Spotfire[12] has a guidance tool called 'View Tip' for rapid assessment of potentially interesting scatterplots, which shows an ordered list of all possible scatterplots from the one with highest correlation to the one with lowest correlation. Guo *et al*.[11,17] also evaluated all possible axis-parallel 2D projections according to the maximum conditional entropy to identify ones that are most useful to find clusters. They visualized the entropy values in a matrix display called the entropy matrix.[24] Our work takes these nascent ideas with the goal of developing a potent framework for discovery.

Rank-by-feature framework

A playful analogy may help clarify our goals. Imagine you are dropped by parachute into an unfamiliar place – it could be a forest, prairie, or mountainous area. You could set out in a random direction to see what is nearby and then decide where to turn next. Or you might go towards peaks or valleys. You might notice interesting rocks, turbulent streams, scented flowers, tall trees, attractive ferns, colorful birds, graceful impalas, and so on. Wandering around might be greatly satisfying if you had no specific goals, but if you needed to survey the land to find your way to safety, catalog the plants to locate candidate pharmaceuticals, or develop a wildlife management strategy, you would need to be more systematic. Of course, each profession that deals with the multifaceted richness of natural landscapes has developed orderly strategies to guide novices, to ensure thorough analyses, to promote comprehensive and consistent reporting, and to facilitate cooperation among professionals.

Our principles for exploratory analysis of multidimensional data sets have similar goals. Instead of wandering, analysts should clarify their goals and use appropriate techniques to ensure a comprehensive analysis. A good starting point is the set of principles put forth by Moore and McCabe, who recommended that researchers should (1) examine each dimension first and then explore relationships among dimensions, and (2) use graphical displays first and then numerical summaries.[25] We extend Moore and McCabe's principles to include ranking the projections to guide discovery of desired features, and realize this idea with overviews to see the range of possibilities and coordination to see multiple presentations. An orderly process of exploration is vital, even though there will inevitably be excursions, iterations,

and shifts of attention from details to overviews and back.

The rank-by-feature framework is especially potent for interactive feature detection in multidimensional data. We use the term, 'features' to include relationships between dimensions (or variables) but also interesting characteristics (e.g. patterns, clusters, gaps, outliers) of the data set. To promote comprehensibility, we concentrate on axis-parallel projections; however, the rank-by-feature framework can be used with general geometric projections. Although 3D projections are sometimes useful to reveal hidden features, they suffer from occlusion and the disorientation brought on by the cognitive burden of navigation. On the other hand, 2D projections are widely understood by users, allowing them to concentrate on the data itself rather than being distracted by navigation controls.

Detecting interesting features in low dimensions (1D or 2D) by utilizing powerful human perceptual abilities is crucial to understand the original multidimensional data set. Familiar graphical displays such as histograms, scatterplots, and other well-known 2D plots are effective to reveal features including basic summary statistics, and even unexpected features in the data set. There are also many algorithmic or statistical techniques that are especially effective in low-dimensional spaces. While there have been many approaches utilizing such visual displays and low-dimensional techniques, most of them lack a systematic framework that organizes such functionalities to help analysts in their feature detection tasks.

Our GRID principles are designed to enable users to better understand distributions in one (1D) or two dimensions (2D), and then discover relationships, clusters, gaps, outliers, and other features. Users work by viewing graphical presentations (histograms, boxplots, and scatterplots), and then choose a feature detection criterion to rank 1D or 2D axis-parallel projections. By combining information visualization techniques (overview, coordination, and dynamic query) with ranking, summaries and statistical methods users can systematically examine the most important 1D and 2D axis-parallel projections. We summarize the GRID principles as:

(1) study 1D, study 2D, then find features
(2) ranking guides insight, statistics confirm.

Abiding by these principles, the rank-by-feature framework has an interface for 1D projections and a separate one for 2D projections. Users can begin their exploration with the main graphical display – histograms for 1D and scatterplots for 2D – and they can also study numerical summaries for more detail.

The rank-by-feature framework helps users systematically examine low-dimensional (1D or 2D) projections to maximize the benefit of exploratory tools. In this framework, users can select an interesting ranking criterion. Users can rank low-dimensional projections (1D or 2D) of the multidimensional data set according to the strength of the selected feature in the projection. When there are many dimensions, the number of possible projections is too large to investigate by looking for interesting features. The rank-by-feature framework relieves users from such burdens by recommending projections to users in an orderly manner defined by a ranking criterion that users selected. This framework has been implemented in our interactive visualization tool, HCE 3.0 (www.cs.umd.edu/hcil/hce/).[2]

1D histogram ordering

Users begin the exploratory analysis of a multidimensional data set by scrutinizing each dimension (or variable) one by one. Just looking at the distribution of values of a dimension gives them useful insight into the dimension. The most familiar graphical display tools for 1D data are *histograms* and *boxplots*. Histograms graphically reveal the scale and skewness of the data, the number of modes, gaps, and outliers in the data. Boxplots are also excellent tools for understanding the distribution within a dimension. They graphically show the five-number summary (the minimum, the first quartile, the median, the third quartile, and the maximum). These numbers provide users with an informative summary of a dimension's center and spread, and they are the foundation of multidimensional data analysis for deriving a model for the data or for selecting dimensions for effective visualization.

The main display for the rank-by-feature framework for 1D projections shows a combined histogram and boxplot (Figure 2-1). The interface consists of four coordinated parts: *control panel*, *score overview*, *ordered list*, and *histogram browser*. Users can select a ranking criterion from a combo box in the control panel, and then they see the overview of scores for all dimensions in the score overview according to the selected ranking criterion. All dimensions are aligned from top to bottom in the original order, and each dimension is color-coded by the score value. By default, cells of high value have bright blue green colors and cells of low value have bright brown colors. The cell of middle value has the white color. As a value gets closer to the middle value, the color intensity attenuates. Users can change the colors for minimum, middle, and maximum values. The color scale and mapping are shown at the top right corner of the overview (Figure 2-1(b)). Users can easily see the overall pattern of the score distribution, and more importantly they can *preattentively* identify the dimension of the highest/lowest score in this overview. Once they identify an interesting row on the score overview, they can just mouse over the row to view the numerical score value and the name of the dimension is shown in a tooltip window (Figure 2-1(b)).

The mouseover event is also instantaneously relayed to the ordered list and the histogram browser, so that the corresponding list item is highlighted in the ordered list (Figure 2-1(c)) and the corresponding histogram and boxplot are shown in the histogram browser (Figure 2-1(d)).

The score overview, the ordered list, and the histogram browser are interactively coordinated according to the change of the dimension in focus. In other words, a change of dimension in focus in one of the three components leads to the instantaneous change of dimension in focus in the other two components.

In the ordered list, users can see the numerical detail about the distribution of each dimension in an orderly manner. The numerical detail includes the five-number summary of each dimension and the mean and the standard deviation. The numerical score values are also shown at the third column whose background is color-coded using the same color-mapping as in the score overview. While numerical summaries of distributions are very useful, sometimes they are misleading. For example, when there are two peaks in a distribution, neither the median nor the mean explains the center of the distribution. This is one of the cases for which a graphical representation of a distribution (e.g., a histogram) works better. In the histogram browser, users can see the visual representation of the distribution of a dimension at a time. A boxplot is a good graphical representation of the five-number summary, which together with a histogram provides an informative visual description of a dimension's distribution. It is possible to interactively change the dimension in focus just by dragging the item slider attached to the bottom of the histogram.

Since different users may be interested in different features in the data sets, it is desirable to allow users to customize the available set of ranking criteria. However, we have chosen the following ranking criteria that we think fundamental and common for histograms as a starting point, and we have implemented them in HCE:

(1) Normality of the distribution (0 to inf) Many statistical analysis methods such as *t*-test, ANOVA are based on the assumption that the data set is sampled from a Gaussian normal distribution. Therefore, it is useful to know the normality of the data set. Since a distribution can be non-normal due to many different reasons, there are at least 10 statistical tests for normality including Shapiro–Wilk test and Kolmogorov–Smirnov test. We used the omnibus moments test for normality in the current implementation. The test returns two values, skewness (s) and kurtosis (k). Since s is 0 and k is 3 for a standard normal distribution, we calculate $|s|+|k-3|$ to measure how the distribution deviates from the normal distribution and rank variables according to the measure. Users can confirm the ranking result using the histogram browser to gain an understanding of how the distribution shape deviates from the familiar bell-shaped normal curve.

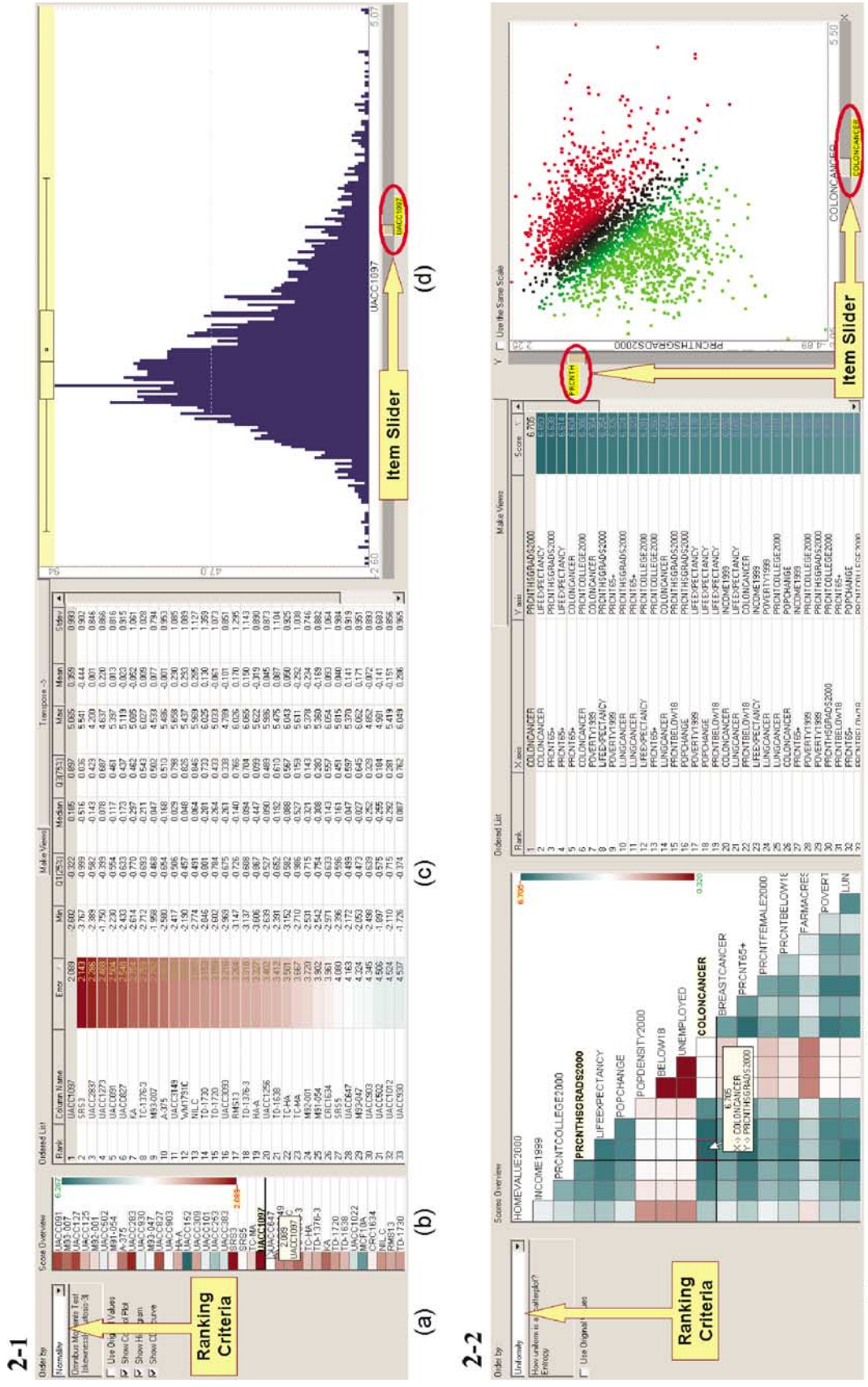
(2) Uniformity of the distribution (0 to number of bins) For the uniformity test, we used an information-based measure called *entropy*. Given k bins in a histogram, the entropy of a histogram H is $entropy(H) = -\sum_{i=1}^{k} p_i \log_2(p_i)$, where p_i is the probability that an item belongs to the ith bin. High entropy means that values of the dimension are from a uniform distribution and the histogram for the dimension tends to be flat. While knowing a distribution is uniform is helpful to understand the data set, it is sometime more informative to know how far a distribution deviates from uniform distribution since a biased distribution sometimes reveals interesting outliers.

(3) The number of potential outliers (0 to number of items) To count outliers in a distribution, we used the $1.5IQR$ (Interquartile range: the difference between the first quartile ($Q1$) and the third quartile ($Q3$)) criterion that is the basis of a rule of thumb in statistics for identifying suspected outliers.[25] An item of value d is considered as a suspected (mild) outlier if $d > (Q3 + 1.5IQR)$ or $d < (Q1 - 1.5IQR)$. To get more restricted outliers (or extreme outliers), $3IQR$ range can be used. It is also possible to use an outlier detection algorithm developed in the data mining. Outliers are one of the most important features not only as noisy signals to be filtered but also as a truly unusual response to a medical treatment worth further investigation or as an indicator of credit card fraud.

(4) The number of unique values (0 to number of items) At the beginning of the data analysis, it is useful to know how many unique values are in the data. A small number of unique values in a large set may indicate problems in sampling or data collection or transcription. Sometimes it may also indicate that the data is a categorical value or the data was quantized. Special treatment may be necessary to deal with categorical or quantized variables.

(5) Size of the biggest gap (0 to max range of dimensions) Gap is an important feature that can reveal separation of data items and modality of the distribution. Let t be a tolerance value, n be the number of bins, and mf be the maximum frequency. We define a gap as a set of contiguous bins $\{b_k\}$ where b_k ($k = 0$ to n) has less than tmf items. The procedure sequentially visits each bin and merges the satisfying bins to form a bigger set of such bins. It is a simple and fast procedure. Among all gaps in the data, we rank histograms by the biggest gap in each histogram. Since we use equal-sized bins, the biggest gap has the most bins satisfying the tolerance value t. It turns out that this ranking criterion is very useful to identify interesting outliers.

For some of the ranking criteria for histogram ordering such as normality, there are many available statistical tests to choose from. We envision that many researchers could contribute statistical tests that could be easily incorporated into the rank-by-feature framework as plug-ins. For example, since outlier detection is a rich research area, novel statistical tests or new data mining algorithms are likely to be proposed in the coming years, and they could be made available as plug-ins.



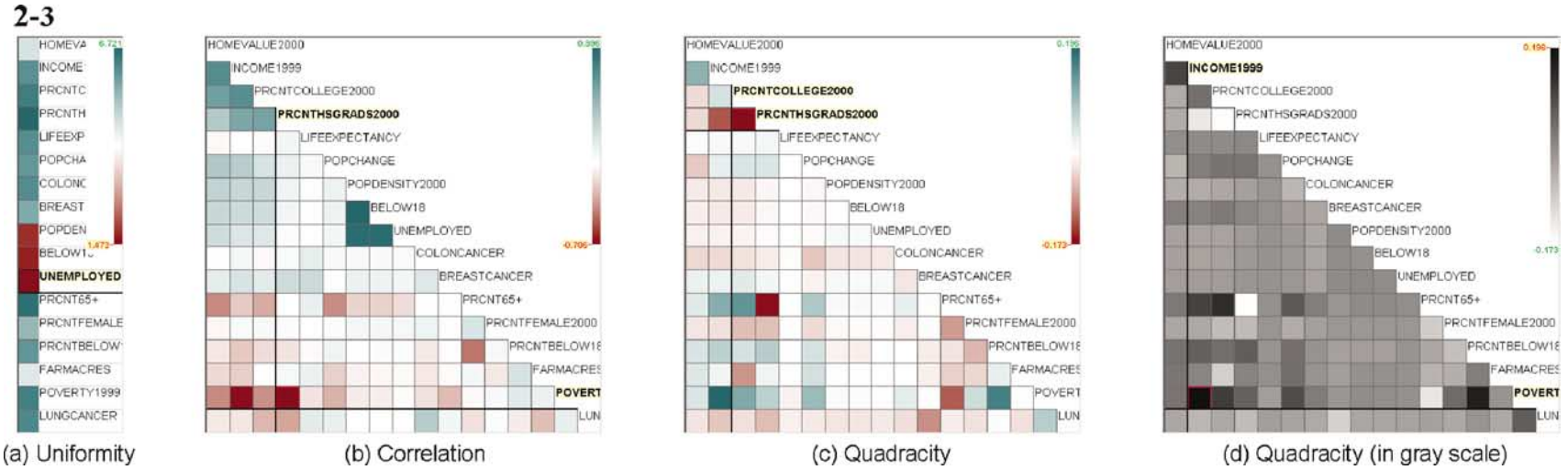


Figure 2 (2-1 and 2-2) Rank-by-feature framework interface for histograms (2-1) and scatterplots (2-2). All 1D histograms and 2D scatterplots are ordered according to the current order criterion (a) in the ordered list (c). The score overview (b) shows an overview of scores of all histograms and scatterplots. Users can select multiple histograms or scatterplots at the same time and generate a separate window for each of them to compare them in a screen. A mouseover event activates a cell in the score overview, highlights the corresponding item in the ordered list (c) and shows the corresponding histogram/scatterplot in the histogram/scatterplot browser (d) simultaneously. A click on a cell selects the cell and the selection is fixed until another click event occurs in the score overview or another selection event occurs in other views. A selected histogram/scatterplot is shown in the histogram/scatterplot browser (d), where users can easily traverse histogram/scatterplot space by changing the dimension using item sliders. Scatterplot browser has two sliders for X- and Y-axes. A boxplot is also displayed above the histogram to show the graphical summary of the distribution of the dimension. The data set shown in 2-1 is a gene expression data set from a melanoma study (3614 genes $\times$ 38 samples), and a demographic and health related statistics for 3,138 U.S. counties with 17 attributes is shown in 2-2. (2-3) The score overviews for U.S. county data. Bright blue green indicates high value and bright brown indicates low value. White is assigned to the value in the middle. When the value varies from negative to positive, white is assigned to the value 0 as in (b). Users who have color deficiencies or who desire different color palettes for their monitors/projectors can change color settings by right clicking on the color scale bar and choosing different colors (d).

2D scatterplot ordering

According to our fundamental principles for improving exploration of multidimensional data, after scrutinizing 1D projections, it is natural to move on to 2D projections where pair-wise relationships will be identified. Relationships between two dimensions (or variables) are conveniently visualized in a scatterplot. The values of one dimension are aligned on the horizontal axis, and the values of the other dimension are aligned on the vertical axis. Each data item in the data set is shown as a point in the scatterplot whose position is determined by the values at the two dimensions. A scatterplot graphically reveals the form (e.g., linear or curved), direction (e.g., positive or negative), and strength (e.g., weak or strong) of relationships between two dimensions. It is also easy to identify outlying items in a scatterplot, but it can suffer from overplotting in which many items are densely packed in one area making it difficult to gauge the density.

We used scatterplots as the main display for the rank-by-feature framework for 2D projections. Figure 2-2 shows the interactive interface design for the rank-by-feature framework for 2D projections. Analogous to the interface for 1D projections, the interface consists of four coordinated components: *control panel*, *score overview*, *ordered list*, and *scatterplot browser*. Users select an ordering criterion in the control panel on the left (Figure 2-2(a)), and then they see the complete ordering of all possible 2D projections according to the selected ordering criterion. The ordered list (Figure 2-2(c)) shows the result of ordering sorted by the ranking (or scores) with scores color-coded on the background. Users can click on any column header to sort the list by the column. Users can easily find scatterplots of the highest/lowest score by changing the sort order to ascending or descending order of score (or rank). It is also easy to examine the scores of all scatterplots with a certain variable for horizontal or vertical axis after sorting the list according to '*X*-axis' or '*Y*-axis' column by clicking the corresponding column header.

However, users cannot see the overview of entire relationships between variables at a glance in the ordered list. Overviews are important because they can show the whole distribution and reveal interesting parts of data. We have implemented a new version of the score overview (Figure 2-2(b)) for 2D projections. It is an *m*-by-*m* grid view where all dimensions are aligned in the rows and columns. Each cell of the score overview represents a scatterplot whose horizontal and vertical axes are dimensions at the corresponding column and row, respectively. Since this grid is symmetric, we used only the lower-triangular part for showing scores and the diagonal cells for showing the dimension names as shown in Figure 2-2(b). Each cell is color-coded by its score value using the same mapping scheme as in 1D ordering. As users move the mouse over a cell, the scatterplot corresponding to the cell is shown in the scatterplot browser (Figure 2-2(d)) simultaneously, and the corresponding item is highlighted in the ordered list (Figure 2-2(c)). Score overview, ordered list, and scatterplot browser are interactively coordinated according to the change of the dimension in focus as in the 1D interface.

In the score overview, users can *preattentively* detect the highest/lowest scored combinations of dimensions thanks to the linear color-coding scheme and the intuitive grid display. Sometimes, users can also easily find a dimension that is the least or most correlated to most of other dimensions by just locating a whole row or column where all cells are the mostly bright brown or bright blue green. It is also possible to find an outlying scatterplot whose cell has distinctive color intensity compared to the rest of the same row or column. After locating an interesting cell, users can click on the cell to select, and then they can scrutinize it on the scatterplot browser and on other tightly coordinated views in HCE.

While the ordered list shows the numerical score values of relationships between two dimensions, the interactive scatterplot browser best displays the relationship graphically. In the scatterplot browser, users can quickly take a look at scatterplots by using item sliders attached to the scatterplot view. Simply by dragging the vertical or horizontal item slider bar, users can change the dimension for the horizontal or vertical axis. With the current version implemented in HCE, users can investigate multiple scatterplots at the same time. They can select several scatterplots in the ordered list by clicking on them with the control key pressed. Then, click 'Make Views' button on the top of the ordered list, and each selected scatterplot is shown in a separate child window. Users can select a group of items by dragging a rubber rectangle over a scatterplot, and the items within the rubber rectangle are highlighted in all other views. On some scatterplots they might gather tightly together, while on other scatterplots they scatter around.

Again interesting ranking criteria might be different from user to user, or from application to application. Initially, we have chosen the following six ranking criteria that we think are fundamental and common for scatterplots, and we have implemented them in HCE. The first three criteria are useful to reveal statistical (linear or quadratic) relationships between two dimensions (or variables), and the next three are useful to find scatterplots of interesting distributions.

(1) Correlation coefficient (−1 to 1) For the first criterion, we use Pearson's correlation coefficient (r) for a scatterplot (S) with n points defined as

$$r(S) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}.$$

Pearson's r is a number between -1 and 1. The sign tells us direction of the relationship and the magnitude tells us the strength of the linear relationship. The magnitude of r increases as the points lie closer to the straight line. Linear relationships are particularly important because

straight line patterns are common and simple to understand. Even though a strong correlation between two variables does not always mean that one variable causes the other, it can provide a good clue to the true cause, which could be another variable. Moreover, dimensionality can be reduced by combining two strongly correlated dimensions, and visualization can be improved by juxtaposing correlated dimensions. As a visual representation of the linear relationship between two variables, the line of best fit or the regression line is drawn over scatterplots.

(2) Least-squares error for curvilinear regression (0 to 1) This criterion is to sort scatterplots in terms of least-squares error from the optimal quadratic curve fit so that users can easily isolate ones where all points are closely/loosely arranged along a quadratic curve. Users are often interested to find nonlinear relationships in the data set in addition to linear relationship. For example, economists might expect that there is a negative linear relationship between county income and poverty, which is easily confirmed by correlation ranking. However, they might be intrigued to discover that there is a quadratic relationship between the two, which can be easily revealed using this criterion.

(3) Quadracity (0 to inf) If two variables show a strong linear relationship, they also produce small error for curvilinear regression because the linear relationship is special cases of the quadratic relationship, where the coefficient of the highest degree term (χ^2) equals zero. To emphasize the real-quadratic relationships, we add 'Quadracity' criterion. It ranks scatterplots according to the coefficient of the highest degree term, so that users can easily identify ones that are more quadratic than others. Of course, the least square error criterion should be considered to find more meaningful quadratic relationships, but users can easily see the error by viewing the fitting curve and points at the scatterplot browser.

(4) The number of potential outliers (0 to number of items) Even though there is a simple statistical rule of thumb for identifying suspected outliers in 1D, there is no simple counterpart for 2D cases. Instead, there are many outlier detection algorithms developed by data mining and database researchers. Among them, distance-based outlier detection methods such as DB-out[26] define an object as an outlier if at least a fraction p of the objects in the data set are apart from the object more than at a distance greater than a threshold value. Density-based outlier detection methods such as LOF-based method[27] define an object as an outlier if the relative density in the local neighborhood of the object is less than a threshold, in other words the local outlier factor (LOF) of the object is greater than a threshold. Since the LOF-based method is more flexible and dynamic in terms of the outlier definition and detection, we included the LOF-based method in the current implementation.

(5) The number of items in the region of interest (0 to number of items) This criterion is the most interactive since it requires users to specify a (rectangular, elliptical, or free-formed) region of interest by dragging the left mouse button on the scatterplot browser. Then the algorithm uses the number of items in the region to order all scatterplots so that users can easily find ones with the most/least number of items in the given 2D region. An interesting application of this ranking criterion is when users specify an upper left or lower right corner of the scatterplot. Users can easily identify scatterplots where most/least items have low value for one variable (e.g., salary of a baseball player) and high value for the other variable (e.g. the batting average).

(6) Uniformity of scatterplots (0 to number of cells) For this criterion, we calculate the entropy in the same way as we did for histograms, but this time we divide the 2D space into regular grid cells and then use each cell as a bin. For example, if we have generated k-by-k grid, the entropy of a scatterplot S is $entropy(S) = -\sum_{i=1}^{k}\sum_{j=1}^{k} p_{ij}\log_2(p_{ij})$, where p_{ij} is the probability that an item belongs to the cell at (i, j) of the grid.

Transformations and potential ranking criteria

Users sometimes want to transform the variable to get a better result. For example, log transformations convert exponential relationships to linear relationships, straighten skewed distributions, and reduce the variance. If variables have differing ranges, then comparisons must be done carefully to prevent misleading results, for example, a gap in a variable whose range is 0–1000 is not usually comparable to a gap in a variable whose range is 2–6. Therefore transformations, such as standardization to common scales, are helpful to ensure that the ranking results are useful. In the current rank-by-feature framework, users can perform five transformations (natural log, standardization, normalization to the first column or to median, and linear scaling to a certain range) over each column or row of the data set when loading the data set. Then when they use the rank-by-feature framework, the results will apply to the transformed values. An improvement to the rank-by-feature framework would allow users to apply transformations during their analyses, not only at the data loading time. More transformations, such as polynomial or sinusoidal functions, would also be useful.

We have implemented only a small fraction of possible ranking criteria in the current implementation. Among the many useful ranking criteria, we suggest three interesting and potent ones.

Modality

If a distribution is normal, there should be one peak in a histogram. However, sometimes there are several peaks. In those cases, different analysis methods (such as sinusoidal fitting) should be applied to the variable, or the dimension should be partitioned to separate each peak (bell-shaped curve). In this sense, the modality is

also an important feature. One possible score for the detection of multimodality could be the change of sign of the first derivative of the histogram curve. If there is one peak, there should be no change at the sign of the first derivative. If there are two peaks, the sign should change once.

Outlierness

The number of outliers can be one of the informative features that contribute to making a better sense of underlying data sets. However, sometimes 'outlierness,' the strength of the outliers in a projection is more informative feature than the number of outliers. The strongest outlier by itself can be a very important signal to users, and at the same time the axes of the projection where the outlier turns out to be a strong outlier can also be informative features because variables for those axes can give an explanation of the outlier's strength. One possible score for the outlierness could be the maximum value of the LOF on a projection.

Gaps in 2D

As we already saw in the 1D ordering cases, gaps are an informative feature in the data set. Several researchers in other fields also have studied related problems such as the largest empty rectangle problem[28,29] and hole detection.[30] The largest empty rectangle problem is defined as follows: Given a 2D rectangular space and points inside it, find the largest axis parallel subrectangle that lies within the rectangle and contains no points inside it. The hole detection problem is to find informative empty regions in a multidimensional space. The time complexity of the current implementations prevents exploratory data analysis. A more rapid algorithm could apply the grid-based approach that was effective in the uniformity criteria. The projection plane can be divided into a relatively small number of grid cells (say 100 by 100), so that it becomes easy to find the biggest gap, similar to the method used for ranking 1D histogram gaps.

Application example

U.S. counties data set

We show an application example of the rank-by-feature framework with a collection of county information data set. The data set has 3,139 rows (U.S. counties) and 17 columns (attributes). In all, 17 attributes are explained in Table 1.

Users first select the 'Uniformity' for 1D ranking, and can preattentively identify the three dimensions ('population,' 'percent under 18 years old,' and 'person unemployed') that have low values in the score overview as shown in Figure 2-3(a). This means the distribution of values of these dimensions is biased to a small range as shown in Figure 3-1(d). The county with the extreme value (highlighted in red at the right most bin of the histogram) on all three low-scored dimensions is 'Los Angeles, CA.' In the histogram for 'percent of high school graduates' that has a high score (Figure 3-1(a)), LA is mapped to a bin below the first quartile on the histogram (also highlighted in red), which means there are relatively lower percentage of high school graduates in LA.

Figure 3-2 shows four histograms ranked by the biggest gap size. Gap detection was performed with standardized values (that is, in this case all dimensions are transformed to a distribution whose mean is 0 and the standard deviation is 1). As discussed in section 4 (opening paragraph), the gap ranking criterion is affected by whether the original or transformed values are used for ranking. Ranking computations based on the original values (values before transformation), produce a different

Table 1 Variables in the U.S. census data set

Variable	*Name*	*Description*
1	HomeValue2000	Median value of owner-occupied housing value, 2000
2	Income1999	Per capita money income, 1999
3	Poverty1999	Percent below poverty level, 1999
4	PopDensity2000	Population, 2000
5	PopChange	Population percent change, 4/1/2000–7/1/2001
6	Prcnt65+	Population 65 years old and over, 2000
7	Below18	Person under 18 years old, 2000
8	PrcntFemale2000	Percent of female persons, 2000
9	PrcntHSgrads2000	Percent of high school graduates age 25+, 2000
10	PrcntCollege2000	Percent of college graduates or higher age 25+, 2000
11	Unemployed	Person unemployed, 1999
12	PrcntBelow18	Percent under 18 years old, 2000
12	LifeExpectancy	Life expectancy, 1997
14	FarmAcres	Farm land (acres), 1997
15	LungCancer	Lung cancer mortality rate per 100,000, 1997
16	ColonCancer	Colon cancer rate per 100,000, 1997
17	BreastCancer	Breast cancer per 100,000 white female, 1994–1997

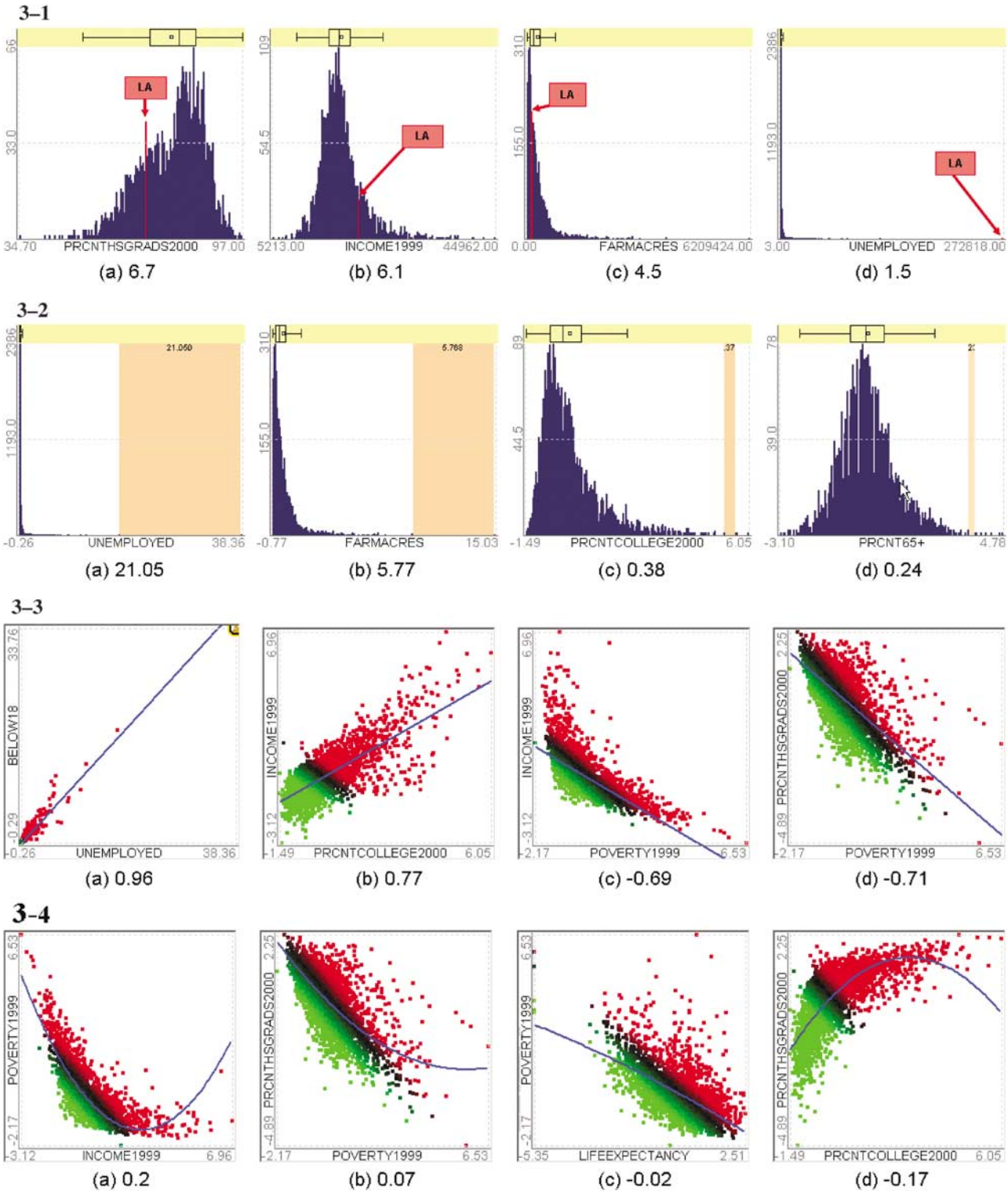


Figure 3 (3-1) Four selected histograms ranging from high uniformity (a) to low uniformity (d). The bar containing Los Angeles, CA (LA) is highlighted in red in each figure. In (d) the distribution is concentrated on the far left and LA appears as an outlier at the far right. (3-2) Four selected histograms ranging from big gap (a) to small gap (d). Gap detection was performed after standardizing each variable. The biggest gap is highlighted as a peach rectangle on each histogram. The bar to the right of the gap on (a) is for LA, and the bar to the right of the gap on (b) is for Coconino, AZ. (3-3) Four selected scatterplots ordered by correlation coefficient. The line of best fit is drawn as a blue line. (3-4) Quadracity (the coefficient of x^2 term). The regression curve is drawn as a blue parabola.

ranking result since the range of the values may change due to the transformation. The biggest gap is highlighted as a peach rectangle on each histogram. The bar to the right of the gap on (a) is for Los Angeles, CA, which confirms the previous ranking result (Figure 3-1(d)). The bar to the right of the gap on (Figure 3-1(b)) is for Coconino, AZ, which means that Coconino County has exceptionally broad farm lands.

Next, if users move on to the rank-by-feature framework for 2D projections, they can choose 'Correlation coefficient' as the ranking criterion. And again they preattentively identify three very bright blue green cells and two very bright brown cells in the score overview (Figure 2-3(b)). The scatterplot for one of the high-scored cells is shown in Figure 3-3(a), where LA is highlighted with an orange triangle in a circle at the top right corner. Interestingly, the three bright cells are composed by the three dimensions that have very low scores in 1D ranking by 'Uniformity.' LA is also a distinctive outlier in all three high scored scatterplots. Users can confirm a trivial relationship between poverty and income, that is poor counties have less income (Figure 3-3(c)). The scatterplot for one of the two bright brown cells is shown in Figure 3-3(d), revealing that counties with high percentages of high school graduates are particularly free from poverty.

User can then run the ranking by quadracity to identify strong quadratic relationships, revealing interesting scatterplots. The score overview for this case is shown in Figure 2-3(c) and (d). Figure 3-4(a) and (d) show weak quadratic relationships. It is interesting to know that they showed strong linear relationships according to the correlation coefficient ranking, but each pair of variables in (a) and (d) also have some weak quadratic relationship. (b) and (c) show almost no quadracity. The fitting errors should be considered by looking into the regression curve and points distribution before confirming the relationships.

Figure 4-1 shows the ranking result using LOF-based outlier detection method. Since the current implementation does not take into account the number of items mapped to the same coordinate, the result is not so accurate, but it still makes sense at most cases. In this ranking result, while it is interesting to know which one has the most outliers, sometimes strong outliers can be found on a scatterplot with the fewest outliers. Future implementations of 'outlierness' could play a better role for this case, for example, Figure 4-1(d) has one strong outlier, Union, FL, where there are a distinctively large number of lung cancer cases and the county is relatively poor.

The rank-by-feature framework is to HCE users what maps are to the explorer of unknown areas. It helps users get some idea about where to turn for the next step of their exploratory analysis of a multidimensional data set. The rank-by-feature framework in HCE 3.0 can handle much larger data sets with many more dimensions than this application example. More columns with environmental, educational, demographic, and medical statistics can be added to this example data set to discover interesting relationships among attributes across many knowledge domains.

A microarray data set

Microarray technology is actively used these days to study gene products. Biologists take samples and hybridize them in gene chips (or microarrays) to measure the activity of genes in the samples. A microarray chip can measure several thousands to tens of thousands of genes. A microarray data set consists of tens or hundreds of microarray chip measurements, so microarray data sets are usually multidimensional. In this section, we show an application example of the rank-by-feature framework with a microarray data set. A group of biologists in the Children's National Medical Center injected a toxin to a murine muscle to examine the process of muscle regeneration process. They took samples from the area where the toxin was injected at 27 different time points and measured the activities of about 12,000 genes.

The biologists start exploring the data set by looking at all 1D projections (or histograms). They can quickly browse all histograms by dragging the item slider in the histogram browser. They easily get to know that all dimensions have a similar distribution that looks like Figure 4-2. In an attempt to rank histograms by the size of the biggest gap, the sample taken at the 16th day (labeled 16D in Figure 4-2) has the biggest gap. These users can select the bar to the right of the gap and learn that the gene name belonging to the bar is 'Troponin T3.' Troponin T3 is related to the muscle contraction. Using the profile search tab in HCE, it turns out that Troponin T3 shows a temporal pattern almost opposite to a candidate gene (MyoD) that is well-known to be related to the muscle regeneration process. These data indicate that further examination of Troponin T3 is warranted to understand how it is related to the muscle regeneration process.

Users move on to the scatterplot ordering tab and try a ranking by correlation coefficient since it is one of the most fundamental and important binary relationships. Figure 4-3 shows the score overview and two scatterplots. The time points are arranged in the sequential order from left to right and from top to bottom in the score overview. By the triangle-shaped blue green squares group (highlighted with a black triangle) in the middle of the overview, users can preattentively perceive that most of time points in the middle are highly correlated to each other as shown in the scatterplot next to the score overview. Similarly, by the rectangular brown squares group (highlighted with a black rectangle) at the bottom left corner, it is easy to know that day 1 (1D) through day 4 (4D) samples do not correlated to the time points at the end (day 16 through day 40). At the same time the brown stripe (highlighted with a black rectangle) at the first column shows that the day 1 through day 4 samples are not correlated to the beginning time point.

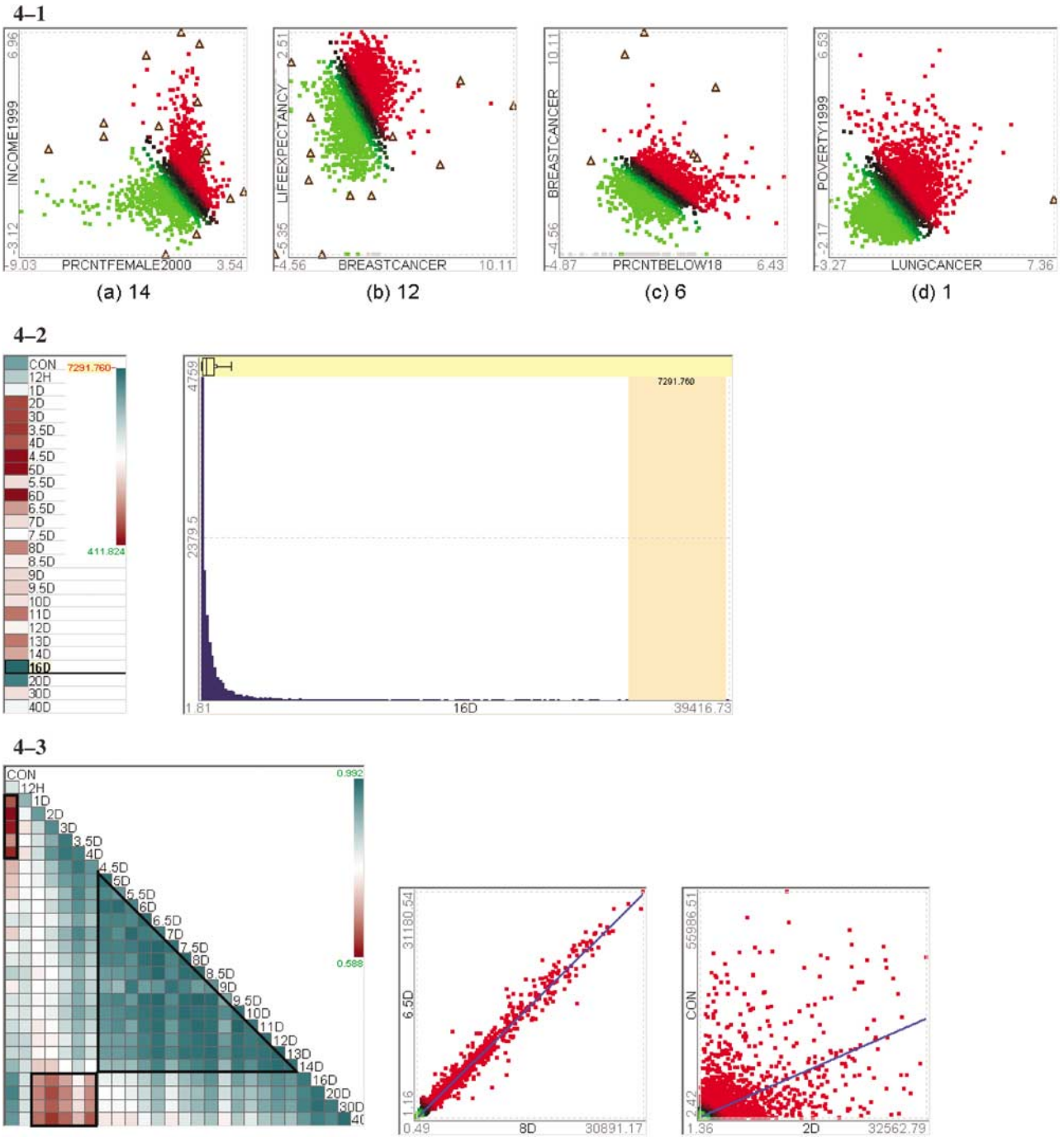


Figure 4 (4-1) The number of outliers. Outliers whose LOF is greater than (minimum LOF + maximum LOF)/2 are highlighted as triangles. (4-2) The ranking result by the size of the biggest gap. The score overview and the top ranked histogram. (4-3) The ranking result by correlation coefficient. The score overview and the top- and bottom-ranked scatterplots.

The rank-by-feature framework saves biostatisticians a significant amount of time to explore the data set by providing efficient graphical summaries and by enabling them to interactively traverse numerous low-dimensional projections. The rank-by-feature framework sometimes leads users to unexpected finding such as distinctive outliers.

Discussion

In spite of their limitations, low-dimensional projections are useful tools for users to understand multidimensional data sets. Since 3D projections have the problem of the cognitive burdens of occlusion and navigation controls, we concentrate on 1D and 2D projections. Since the axis-parallel projections are much more easily interpreted by

Table 2 Computation times (in seconds) to complete 2D rankings for four data sets of various sizes (# of items by # of dimensions)

Size	*Criterion*			
	Correlation	*Curvilinear regression and quadracity*	*Uniformity*	*Number of outliers (LOF)*
3138×17	0.05	0.2	0.2	4.1
3614×38	0.1	0.8	1.6	39.0
11704×105	2.6	17.4	38.6	810.2
22283×105	4.9	33.1	72.5	1660.0

users compared to arbitrary 1D or 2D projections, we concentrate on axis-parallel 1D and 2D projections.

The rank-by-feature framework supports comprehensive exploration of these axis-parallel projections. Interactive interfaces for the rank-by-feature framework were designed for 1D and 2D projections. There are four coordinated components in each interface: control panel, score overview, ordered list, and histogram/scatterplot browser. Users choose a ranking criterion at the control panel, and then they can examine the ranked result using the remaining three coordinated components. The score overview enables users to preattentively spot distinctive high and low ranked projections due to the consistent layout and linear color-mapping, and it also helps users to grasp the overall pattern of the score distribution. While the ordered list provides users with the numerical summary of each projection, the browser enables users to interactively examine the graphical representation of a projection (the combination of histogram and boxplot for a 1D projection, and scatterplot for a 2D projection). The item slider attached to histogram/scatterplot display facilitates the exploration by allowing the interactive change of the dimension in focus.

When implementing or selecting a new ranking criterion for the rank-by-feature framework, implementers should strive to limit the time complexity of the score function for the criterion. If there are n data items in m-dimensional space, the score function of a 2D projection is calculated $m(m\text{-}1)/2$ times. If the time complexity of the score function is $O(n)$, the total time complexity will be $O(nm^2)$. Reasonable response times are achievable if there are efficient algorithms for computing scores for a ranking criterion. Otherwise, it is necessary to develop a quickly computable approximate measure in order to cut down the processing time. A grid cell-based approach can reduce the response time by running the algorithm on a smaller number of cells instead of actual data points. Table 2 shows the amount of CPU time (in seconds) to complete 2D rankings for four data sets of various sizes (# of items by # of dimensions) with our current implementation on a Intel Pentium 4 (2.53 GHz, 1 GB memory) PC running a Windows XP Professional operating system.

In terms of scalability, the score overview is certainly better than the scatterplot matrix where a small thumbnail of the actual scatterplot is shown in each cell. However, when there are too many dimensions, the score overview will become so crowded that it will be difficult to view and to read the labels. Since the screen space should be shared with other views, the score overview becomes unacceptably overcrowded in a general PC environment when the dimensionality is greater than about 130. In that case, a filtering or grouping mechanism will be necessary. A range slider to the right side of the score overview might control the upper and lower bound of scores displayed. If the score of a cell does not satisfy the thresholds, the cell will be grayed out. If an entire row or column is grayed out, the row or column can be filtered out so that remaining rows and columns will occupy more screen space. Implementers can also utilize the dimension (or column) clustering result that is in HCE to rank clusters of dimensions instead of individual dimensions.

Conclusion

The take-away message from the natural landscape analogy in the earlier section on the rank-by-feature framework is that guiding principles can produce an orderly and comprehensive strategy with clear goals. Even when researchers are doing exploratory data analysis, they are more likely to make valuable insights if they have some notion of what they are looking for. There are lots of creatures (and features) hiding in high-dimensional spaces, so researchers and data analysts will do better if they decide whether they are looking for birds, cats, or fish.

We believe that our proposed strategy for multidimensional data exploration with room for iteration and rapid shifts of attention enables novices and experts to make discoveries more reliably. The GRID principles are:

(1) study 1D, study 2D, then find features
(2) ranking guides insight, statistics confirm.

The rank-by-feature framework enables users to apply a systematic approach to understanding the dimensions and finding important features using axis-parallel 1D and 2D projections of multidimensional data sets. Users begin

by selecting a ranking criterion and then can see the ranking for all 1D or 2D projections. They can select high- or low-ranked projections and view them rapidly, or sweep through a group of projections in an orderly manner. The score overview provides a visual summary that helps users identify extreme values of criteria such as correlation coefficients or uniformity measures. Information visualization principles and techniques such as dynamic query by item sliders, combined with traditional graphical displays such as histograms, boxplots, and scatterplots play a major role in the rank-by-feature framework.

As future work, various statistical tools and data mining algorithms, including ones presented in the next section, can be incorporated into our rank-by-feature framework as new ranking criteria. Just as geologists, naturalists, and botanists depend on many kinds of maps, compasses, binoculars, or Global Positioning Systems, dozens of criteria seem useful in our projects. It seems likely that specialized criteria will be developed by experts in knowledge domains such as genomics, demographics, and finance. Other directions for future work include extending the rank-by-feature framework to accommodate 3D projections and generalizing to categorical and binary data.

We recognize that the concepts in the rank-by-feature framework and the current user interface will be difficult for many data analysts to master. However, our experience in gene expression data analysis tasks and with a dozen biologists is giving us a better understanding of what training methods to use. Of particular importance is the development of meaningful examples based on comprehensible data sets that demonstrate the power of each ranking criterion. Screen space is a scarce resource in these information abundant interfaces, so higher resolution displays (we use $3{,}800 \times 2{,}480$ pixel display whenever possible) or multiple display are helpful, as are efficient screen management strategies.

User studies may help us improve the user interface, but the central contributions of this paper are the potent concepts in the rank-by-feature framework. We hope they will be implemented by others with varied interfaces for spreadsheets, statistical packages, or information visualization tools. We believe that the GRID principles and the rank-by-feature framework will effectively guide users to understand dimensions, identify relationships, and discover interesting features.

Acknowledgments

This work was supported by N01 NS-1-2339 from the NIH and by the National Science Foundation under Grant No. EIA 0129978. We thank IBM for their gift of the large display. We also thank Ben Bederson, Di Cook, Francois Guimbretiere, Diansheng Guo, Harry Hochheiser, Alan MacEachren, Lee Wilkinson, and peer reviewers for giving us constructive suggestions on revising our paper. We appreciate the support from and partnership with Eric Hoffman and his lab at the Children's National Medical Center, through NIH grant N01-NS-1-2339.

References

1 Inselberg A, Dimsdale B. *Parallel Coordinates: A Tool for Visualizing Multidimensional Geometry*. IEEE Symposium on Information Visualization 1990 (San Francisco, CA, USA) IEEE Computer Society Press: Chicago, 361–375.
2 Seo J, Shneiderman B. Interactively exploring hierarchical clustering results. *IEEE Computer* 2002; **35**: 80–86.
3 Seo J, Shneiderman B. *A Rank-by-feature Framework for Unsupervised Multidimensional Data Exploration using Low-dimensional Projections*. IEEE Symposium on Information Visualization 2004 (Austin, TX, USA), IEEE Computer Society Press: Chicago, 65–72.
4 Kohonen T. *Self-Organizing Maps*. 3rd edn. Springer: New York, 2000.
5 Friedman JH. Exploratory projection pursuit. *Journal of American Statistical Assocication* 1987; **82**: 249–266.
6 Friedman JH, Tukey JW. A projection pursuit algorithm for exploratory data analysis. *IEEE Transaction on Computer* 1974; **23**: 881–890.
7 Asimov D. The grand tour: a tool for viewing multidimensional data. *The SIAM Journal of Scientific and Statistical Computing* 1985; **6**: 128–143.
8 Huber PJ. Projection Pursuit. *The Annals of Statistics* 1985; **13**: 435–475.
9 Cook D, Buja A, Cabrera J, Hurley C. Grand Tour and Projection Pursuit. *Journal of Computational and Graphical Statistics* 1995; **4**: 155–172.
10 Hinneburg A, Keim DA, Wawryniuk M. HD-Eye: Visual mining of high-dimensional data. *IEEE Computer Graphics and Applications* 1999; **19**: 22–31.
11 Guo D. Coordinating computational and visual approaches for interactive feature selection and multivariate clustering. *Information Visualization* 2003; **2**: 232–246.
12 Spotfire DecisionSite, Spotfire. http://www.spotfire.com/(accessed 21 October 2004).
13 Ward MO. *XmdvTool: Integrating multiple methods for visualizing multivariate data*. IEEE Visualization 1994 (Washington DC, USA), IEEE Computer Society Press: Chicago, 326–336.
14 Liu H, Motoda H. *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic Publishers: Boston, 1998.
15 Aggarwal CC, Procopiuc C, Wolf J, Yu P, Park JS. *Fast Algorithms for Projected Clustering*. ACM SIGMOD Conference 1999 (Philadelphia, USA) ACM Press: New York, 61–72.
16 Agrawal R, Gehrke J, Gunopulos D, Raghavan P. *Automatic subspace clustering of high dimensional data for data mining applications*. ACM SIGMOD Conference 1998 (Seattle, WA, USA), ACM Press: New York, 94–105.
17 Guo D, Gahegan M, Peuquet D, MacEachren A. *Breaking Down Dimensionality: An Effective Feature Selection Method for High-Dimensional Clustering*. The Third SIAM (Society for Industrial and Applied Mathematics) International Conference on Data Mining, Workshop on Clustering High Dimensional Data and its Applications 2003 (San Francisco, CA, USA), 29–42.
18 Bertin J. *Graphics and Graphic Information Processing*. Walter de Gruter & Co.: Berlin, 1981.
19 Ankerst M, Berchtold S, Keim DA. *Similarity Clustering of Dimensions for an Enhanced Visualization of Multidimensional Data*. IEEE Symposium on Information Visualization 1998 (Research Triangle Park, NC, USA), IEEE Computer Society Press: Chicago, 52–60.
20 Friendly M. Corrgrams: exploratory displays for correlation matrices. *The American Statistician* 2002; **19**: 316–325.
21 Yang J, Peng W, Ward MO, Rundensteiner EA. *Interactive Hierarchical Dimension Ordering, Spacing and Filtering for Exploration of High*

Dimensional Datasets. IEEE Symposium on Information Visualization 2003 (Seattle, WA, USA), IEEE Computer Society Press: Chicago, 105–112.
22 Amar R, Stasko J. *A knowledge task-based framework for design and evaluation of information visualizations*. IEEE Symposium on Information Visualization 2004 (Austin, TX, USA), IEEE Computer Society Press: Chicago, 143–149.
23 Tukey JW, Tukey PA. *Computer graphics and exploratory data analysis: An introduction*. Annual Conference and Exposition: Computer Graphics 1985 (Fairfax, VA, USA), Vol. 3, National Micrograhics Association: Silver Spring, 773–785.
24 MacEachren A, Dai X, Hardisty F, Guo D, Lengerich G. *Exploring High-d Spaces with Multiform Matricies and Small Multiples*. IEEE Symposium on Information Visualization 2003 (Seattle, WA, USA), IEEE Computer Society Press: Chicago, 31–38.
25 Moore DS, McCabe GP. *Introduction to the Practice of Statistics*. 3rd edn. W.H. Freeman and Company: New York, 1999.
26 Knorr EM, Ng RT, Tucakov V. Distance-based outliers: algorithms and applications. *The International Journal on Very Large Data Bases* 2000; **8**: 237–253.
27 Breunig MM, Kriegel HS, Ng RT, Sander J. *LOF: identifying density-based local outliers*. ACM SIGMOD Conference 2000 (Dallas, Texas, USA), ACM Press: New York, 93–104.
28 Chazelle B, Drysdale RL, Lee DT. Computing the largest empty rectangle. *The SIAM Journal of Scientific and Statistical Computing* 1986; **15**: 550–555.
29 Edmonds J, Gryz J, Liang D, Miller RJ. Mining for empty spaces in large data sets. *Theoretical Computer Science* 2003; **296**: 435–452.
30 Liu B, Ku LP, Hsu W. *Discovering interesting holes in data*. International Joint Conference on Artificial Intelligence 1997 (Nagoya, Japan), Morgan Kaufmann: San Francisco, 930–935.
31 Swayne DF, Lang DT, Buja A, Cook D. GGobi: evolving from XGobi into an extensible framework for interactive data visualization. *Computional Statistics & Data Analysis* 2003; **43**: 423–444.