TECHNICAL RESEARCH REPORT

A Rank-by-Feature Framework for Interactive Exploration of Multidimensional Data (2004)

by Jinwook Seo, Ben Shneiderman

TR 2005-62

A Rank-by-Feature Framework for Interactive Exploration of Multidimensional Data

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

[1]Department of Computer Science,
[2] Human–Computer Interaction Lab, Institute for Advanced Computer Studies,
[3]Institute for Systems Research

University of Maryland,
College Park, MD 20742 U.S.A

Correspondence:
Jinwook Seo
Department of Computer Science,
A.V. Williams Building
College Park, MD 20742, U.S.A.
Tel: +1 301-405-2725,
Fax: +1 301-405-6707
E-mail: jinwook@cs.umd.edu

A possible running title: A Rank-by-Feature Framework

Acknowledgement: 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.

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) 1D, 2D, then features (2) graphics, ranking, summaries, then statistics. 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.

Keywords: rank-by-feature framework, information visualization, exploratory data analysis, dynamic query, feature detection/selection, graphical displays, summaries, statistical tests.

1 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 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 two-dimensional (2D) and three-dimensional (3D) spaces, and computer displays are intrinsically 2D, 2D projections have been widely used as useful 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.

The three categories of two-dimensional presentations are distinguished 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 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 [16].

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, users can select an interesting ranking criterion, and then all possible axis-parallel projections are ranked by the selected ranking criterion (Figure 1). The ranking result is visually presented in 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 not only easily perceive the most interesting projections, but also grasp the overall ranking score distribution. It is also possible to 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).

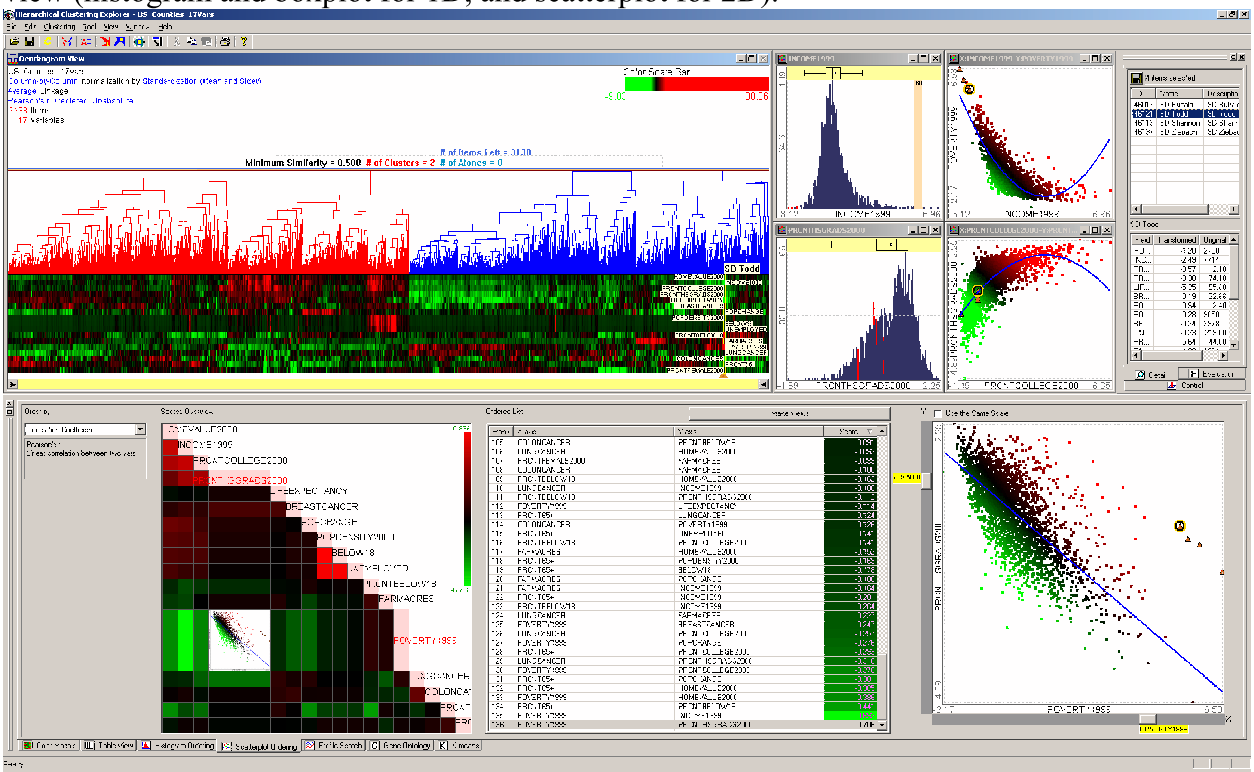

Figure 1. The Hierarchical Clustering Explorer (HCE) The rank-by-feature framework is implemented as two new tab windows in HCE 3.0. The main view is the dendrogram view where users can interactively explore hierarchical clustering results [17]. Whenever users identify an interesting projection in the rank-by-feature framework, they can generate a separate child window that will interactively coordinate with all other views in HCE 3.0. A 2D scatterplot ordering result (section 3.2) by correlation coefficient is shown with the U.S. counties data set (section 5). Two scatterplots and two histograms are shown in separate views together with the clustering result view. Four counties that are poor and have medium number of high school graduates are selected in the scatterplot browser and they are all highlighted in other views (with triangles).

We implemented the rank-by-feature framework in our interactive exploration tool for multidimensional data, the Hierarchical Clustering Explorer (HCE) [17] (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.

Section 2 introduces related work, and section 3 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 section 4. An application example is presented in section 5. Discussion and future work are in section 6. We conclude the paper in section 7.

2 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, multidimensional scaling (MDS), and parallel coordinates [16] are used to find informative two-dimensional projections of multidimensional data sets. Self-organizing maps (SOM) [18] can also be thought of as a projection technique. Taking a look at only a single projection for a multidimensional data is not enough to discover all the interesting features in the original data since any one projection may obscure some features [12]. Thus it is inevitable that users must scrutinize a series of projections to reveal the features of the data set.

Projection methods belonging to category (1), non-axis-parallel, were developed by statisticians. The idea of projection pursuit [12] is to find the most interesting low dimensional projections to identify interesting features in the multidimensional data set. An automatic projection pursuit method, known as the grand tour [5], is a method for viewing multidimensional data via orthogonal projection onto a sequence of two-dimensional subspaces. It changes the viewing direction, generating a movie-like animation that makes a complete search of the original space. However, it might take several hours to complete a reasonably complete visual search in four dimensions [15]. An exhaustive visual search is out of the question as the number of dimensions grows.

Friedman and Tukey [12] devised a method to automate the task of projection pursuit. They defined interesting projections as ones deviating from the normal distribution, and provide a numerical index to indicate the interestingness of the projection. When an interesting projection is found, the features on the projection are extracted and projection pursuit is continued until there is no remaining feature found. XGobi [8] is a widely-used graphical tool that implemented both the grand tour and the projection pursuit, but not the ranking that we propose.

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 [14][22][24].

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 this work, the term, feature selection is a process that chooses an optimal subset of features according to a certain criterion [19], where feature simply means dimension. Basically, the goal is to find a good subset of dimensions (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 [1][2][14]. 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, some researchers have 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 [4][13][25]. Yang et al. [25] proposed innovative dimension ordering methods to improve the effectiveness of visualizations 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 & Stasko [3].

In early 1980's, Tukey [23] envisioned a concept of ranking scatterplots by numerical indices called "Scagnostics." He is one of the prominent statisticians who foresaw the utility of computers in exploratory data analysis. We brought his concept to reality with the rank-by-feature framework in the Hierarchical Clustering Explorer. There are also some research tools and commercial products for helping users to find more informative visualizations. Spotfire [22] 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. [14] 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 [20]. Our work takes these nascent ideas with the goal of developing a potent framework for discovery.

3 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 multi-faceted 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 statistical tools should (1) enable users to examine each dimension first and then explore relationships among dimensions, and (2) offer graphical displays first and then provide numerical summaries [21]. 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 unsupervised multidimensional data. We use the term, "features" to include relationships between dimensions (or variables) but also interesting characteristics (patterns, clusters, gaps, outliers, or items) 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 Graphics, Ranking, and Interaction for Discovery (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 summaries and statistical methods users can systematically examine the most important 1D and 2D axis-parallel projections. We summarize the GRID principles as:

(1) 1D, 2D, then features
(2) graphics, ranking, summaries, then statistics.

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 every one randomly looking for interesting features. The rank-by-feature framework relieves users from such burdens by recommending projections to users in an ordered 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/) [17].

3.1 1D HISTOGRAM ORDERING

Users begin the exploratory analysis of a multidimensional data by scrutinizing each dimension (or variable) one by one. Just looking at the distribution of values of a dimension gives us 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 detecting and illustrating location and variation changes of 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). 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 red colors and cells of low value have bright

green colors. The cell of middle value has the black 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 (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).

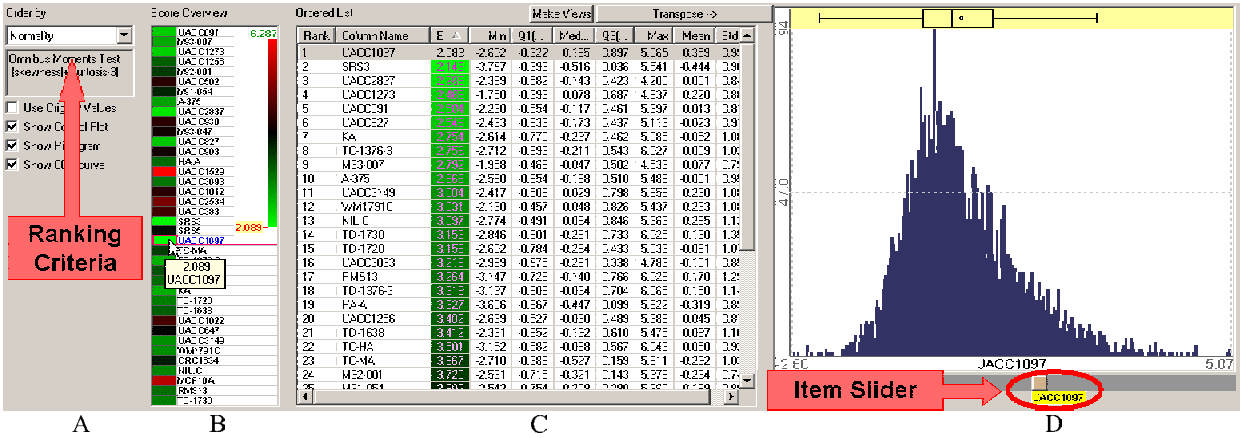

Figure 2. Rank-by-feature framework interface for histograms (1D) All 1D histograms are ordered according to the current order criterion (A) in the ordered list (C). The score overview (B) shows an overview of scores of all histograms. A mouseover event activates a cell in the score overview, highlights the corresponding item in the ordered list (C) and shows the corresponding histogram in the histogram browser (D) simultaneously. A click on a cell selects the cell and the selection is fixed until another click event occurs. A selected histogram is shown in the histogram browser (D), where users can easily traverse histogram space by changing the dimension for the histogram using item slider. A boxplot is also displayed above the histogram to show the graphical summary of the distribution of the dimension. (Data shown is from a gene expression data set from a melanoma study (3614 genes x 38 samples)).

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 and the corresponding histogram and boxplot are shown in the histogram browser. 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 nonnormal due to many different reasons, there are at least ten 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 *inf*):

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 *i*-th 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 *n*):

To count outliers in a distribution, we used the 1.5**IQR* (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 [21]. An item of value *d* is considered as a suspected (mild) outlier if $d > (Q3+1.5*IQR)$ or $d < (Q1-1.5*IQR)$. To get more restricted outliers (or extreme outliers), 3**IQR* 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 *n*)

At the beginning of the data analysis, it is useful to know how many unique values are in the data. Only small number of unique values in a large set may indicate problems in sampling or data collection or transcription. Sometime 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 $t*mf$ 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.

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.

3.2 2D SCATTERPLOT ORDERING

According to our fundamental principles for improving exploration of unsupervised 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.

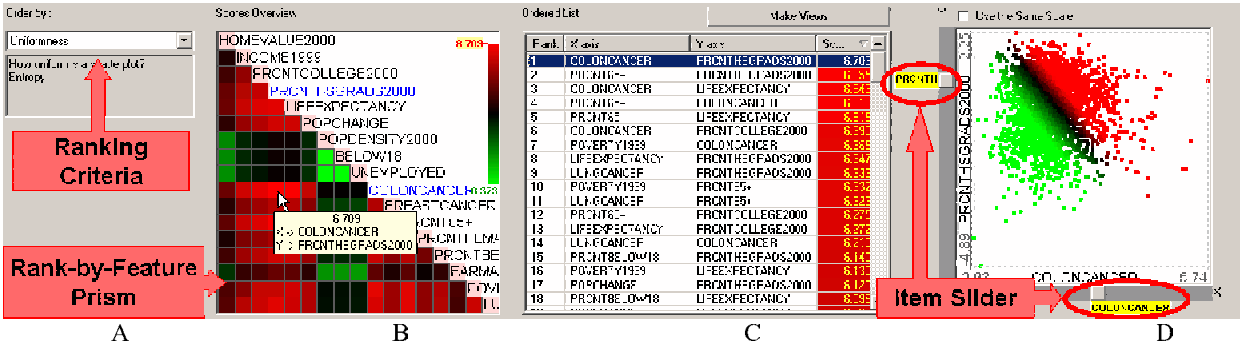

Figure 3. Rank-by-feature framework interface for scatterplots (2D) All 2D scatterplots are ordered according to the current ordering criterion (A) in the ordered list (C). Users can select multiple scatterplots at the same time and generate separate scatterplot windows for them to compare them in a screen. The score overview (B) shows an overview of scores of all scatterplots. Mouseover event activates a cell in the score overview, highlights the corresponding item in the ordered list (C) and shows the corresponding scatterplot in the scatterplot browser (D) simultaneously. A click on a cell selects the cell and the selection is fixed until another click event occurs. A selected scatterplot is shown in the scatterplot browser (D), where it is also easy to traverse scatterplot space by changing X or Y axis using item sliders on the horizontal or vertical axis. (A demographic and health related statistics for 3138 U.S. counties with 17 attributes.)

We used scatterplots as the main display for the rank-by-feature framework for 2D projections. Figure 3 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, and then they see the complete ordering of all possible 2D projections according to the selected ordering criterion (Figure 3A). The ordered list 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 or Y 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 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 table 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 3B. 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 simultaneously, and the corresponding item is highlighted in the ordered list (Figure 3C). 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 green or bright red. 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 doesn't 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 square error for curvilinear regression (0 to 1)

This criterion is to sort scatterplots in terms of least-square errors 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 (x^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 n)

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 [10] 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 [6] 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 *n*)

This criterion is the most interactive since it requires users to specify a (rectangular, elliptical, or free-formed) region of interest. Then the algorithm uses the number of items in the region to order scatterplots so that users can easily find ones with most/least number of items in the given region. An interesting application of this ranking criterion is when a user specifies 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 *inf*)

For this criterion, we calculate the entropy in the same way as we did for histograms, but this time we divide the two-dimensional 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.

4 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, e.g. 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 5 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.