Visual Correlation Analysis of Numerical and Categorical Data on the Correlation Map

Zhiyuan Zhang, Kevin T. McDonnell,
Erez Zadok, *Member, IEEE*, and Klaus Mueller, *Senior Member, IEEE*

Abstract—Correlation analysis can reveal the complex relationships that often exist among the variables in multivariate data. However, as the number of variables grows, it can be difficult to gain a good understanding of the correlation landscape and important intricate relationships might be missed. We previously introduced a technique that arranged the variables into a 2D layout, encoding their pairwise correlations. We then used this layout as a network for the interactive ordering of axes in parallel coordinate displays. Our current work expresses the layout as a correlation map and employs it for visual correlation analysis. In contrast to matrix displays where correlations are indicated at intersections of rows and columns, our map conveys correlations by spatial proximity which is more direct and more focused on the variables in play. We make the following new contributions, some unique to our map: (1) we devise mechanisms that handle both categorical and numerical variables within a unified framework, (2) we achieve scalability for large numbers of variables via a multi-scale semantic zooming approach, (3) we provide interactive techniques for exploring the impact of value bracketing on correlations, and (4) we visualize data relations within the sub-spaces spanned by correlated variables by projecting the data into a corresponding tessellation of the map.

Index Terms—Visual analytics, Visual correlation analysis, Categorical data, Information visualization, Interactive interfaces

1 Introduction

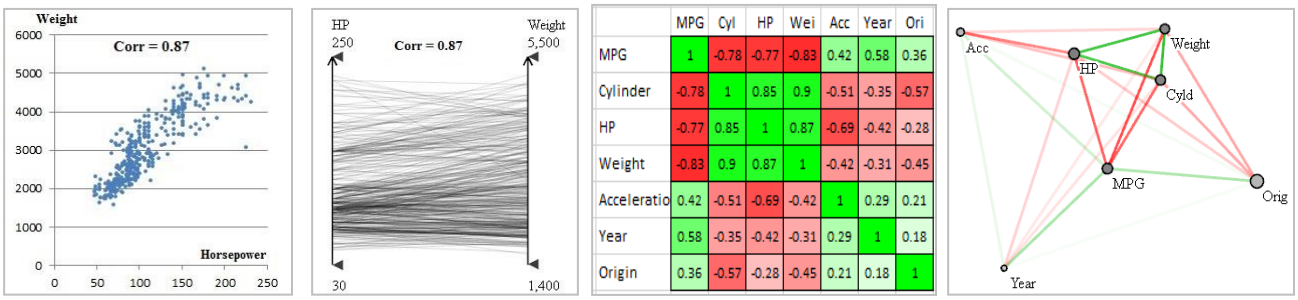
The rapid development of information technology produces vast amounts of data with numerous attributes. These high-dimensional datasets offer tremendous opportunities for studying behavioral patterns and also for predicting future developments. Valuable insight often comes from intricate inter-relationships that exist among data attributes (or *variables*). For example, in psychology research, scientists try to find relationships between intelligence, aptitude and social behavior. In finance and economics, to maximize profit, economists look for the group of variables that are mostly related to profit. Finally, in the social and natural sciences, researchers seek to understand and explain the nature of relations between certain phenomena. To make progress in this wide gamut of areas, analysts require effective, sensitive, and intuitive tools to uncover these relationships.

Correlation analysis is one such tool. It looks for relationships between variables and can show whether pairs of variables are related and how strongly. Correlation analysis has become increasingly popular in many fields, including psychology, education, finance, marketing, and climatology, just to name a few. Correlations, however, are difficult to interpret, manage, and survey once the number of variables becomes even moderately large. Given D variables, there are $O(D^2)$ correlation pairs, which makes complex relationships difficult to recognize from columns of numbers alone. Hence, there is a clear need for an effective visual interface that allows analysts to (1) quickly get an overview of the overall correlation relationships in the data, and (2) easily manipulate the data to reveal hidden relationships via different modes of interactions, such as filtering, selection, bracketing, and clustering.

Correlation analysis is related to regression analysis. While regression analysis quantifies the linear relationship between a dependent variable and one or more independent variables, correlation analysis makes no distinction between independent and dependent variables – it is only a measure of linear association between two variables. The strength of this linear association is gauged by the *correlation coefficient r*. It is this coefficient that links correlation and regression analysis, because squaring r, or r^2, yields the *coefficient of determination* which is a measure of how well the regression line represents the data. It is important to realize, however, that neither correlation nor regression analysis can establish cause-and-effect relationships among the variables. These can only be inferred by a human analyst, and this circumstance forms a main motivation of our work.

- *Zhiyuan Zhang and Klaus Mueller are with the Visual Analytics and Imaging Lab at the Computer Science Department, Stony Brook University, Stony Brook, NY. Email: {zyzhang, mueller}@cs.sunysb.edu.*
- *Klaus Mueller is also with the Computer Science Dept., SUNY Korea*
- *Kevin T. McDonnell is with the Department of Mathematics and Computer Science, Dowling College, Oakdale, NY. E-mail: mcdonnek@dowling.edu.*
- *Erez Zadok is with the Filesystems and Storage Lab at the Computer Science Department, Stony Brook University, Stony Brook, NY. Email: ezk@cs.stonybrook.edu*

Manuscript received (insert date of submission if desired). Please note that all acknowledgments should be placed at the end of the paper, before the bibliography.



(a) (b) (c) (d)

Fig. 1. Examples of different correlation representations: (a) scatterplots of two variables, (b) parallel coordinates plot of the same two variables., (c) correlation matrix display of 7 variables with color and intensity coding correlation sign and strength, (d) our correlation map.

Since we consider the variables the main actors, we seek a visual interface that can best show how variables – categorical and numerical – interact with one another, using spatial proximity encoding to convey the strength of these interactions. We call this visual interface the *Correlation Map*. It builds on our prior effort reported in [33] where we first applied a force-directed algorithm to optimize a correlation-centric 2D layout of all variables and then computed a low-cost path across it to determine a good ordering of axes in a parallel coordinate display. Users could manipulate this path directly in the interface and so modify the axes orderings intuitively. Our present work also uses this layout but it does so for a different purpose – interactive visual correlation analysis. To enable this we have introduced a new set of interactions and visual representations. More concretely, we have devised:

- new mechanisms that can handle both categorical and numerical variables within a unified framework
- a multi-scale zooming approach to achieve scalability for large numbers of variables
- interactive techniques for exploring the impact of value bracketing on correlations
- a tiled visualization of the subspaces spanned by sets of correlated variables

Our article is organized as follows. Section 2 discusses related work and provides a contrast with our own. Section 3 presents theoretical aspects. Section 4 introduces our algorithm for transforming categorical to numerical data. Section 5 describes our visual analytics framework. Section 6 reports on applications. Section 7 presents a discussion. Section 8 ends with conclusions.

2 Related Work

2.1 Correlation Visualization – A View in Contrast

Correlation is visually expressed as patterns the data form in the display. In 2D scatterplots, the more closely the point cloud adheres to a straight line, the greater the correlation of the two variables (Fig. 1a). On the other hand, in a parallel coordinate display [13] a strong positive correlation between two variables is visually expressed by coherent bundles of lines with similar slope (Fig. 1b). Likewise, a strong negative correlation causes these coherent bundles to form a classic bowtie shape with a focused cross-over point. The findings by Li et al. [15] (and others cited in Li et al.), however, indicate that both representations suffer from inaccurate assessment of correlations. The inaccuracies emerge because the viewer is asked to map a visual pattern to a numerical quantity – the correlation factor. This requires training in pattern recognition, falls victim to non-linear perceptual processes, and is thus prone to human judgment errors.

At the same time, both 2D scatterplots and parallel coordinates share one further shortcoming – they cannot easily show correlations that involve more than two variables. Although 2D scatterplots can be expanded into scatterplot matrices (SPLOM) [9] and the axes in parallel coordinates can be re-ordered [7][19] to expose relationships of different sets of variables, integrating this disparate information across either tiles or axes is difficult.

An alternative approach is to visualize the correlation matrix directly, providing a holistic view over the variable space (Fig 1c). In this representation, each matrix cell denotes the correlation of one variable pair. The matrix view has found a wide set of applications. Seo and Shneiderman [24] use a matrix-based visualizer to provide an overview of the ranking of features, while Henry and Fekete [10] integrate the node-link diagram with a matrix-based display to support the exploration of social networks. Many of these methods support interactive filtering and clustering, and also matrix reordering [26]. In our case it can reveal clusters of correlated variables.

The perception of the clusters can be further enhanced by coding correlation strength to color, yielding a heatmap [24]. However, according to Bertin's levels of organization [2], brightness and color are poor visual variables for quantitative information – spatial variables such as size and proximity are far better choices.

It is for these various reasons why we prefer spatial representations over color, brightness, and data patterns to encode correlation strength and other statistical information about data variables (Fig. 1d). Whereas Ghoniem et al. [8] find that matrix representations outperform graphs when the number of nodes exceeds 20, our application only minimally shares the tasks they test, mainly because our representation is not a graph, but a map. We also reduce edge and node complexity by interactive thresholding, filtering, and level-of-detail management.

The notion of correlation has been widely used in information visualization and visual analytics. Yang et al. [32] present the Value and Relation (VaR) framework that allows users to explore large datasets with large numbers

of dimensions. In their work, glyphs are used to represent values in dimensions, and the locations of the glyphs reveal relationships among dimensions. Qu et al. [20] visualize correlated dimensions in terms of a network layout in which the relative distance of vertices encodes strength of correlation, but no facilities for visual correlation analysis are presented. Chen et al. [4] and Sukharev et al. [28] utilize visualization techniques to show correlations in time-varying multivariate climate datasets with 3D spatial references. More recently, also for time-varying multivariate data, Biswas et al. [3] used mutual information and information overlap as correlation. They utilized our layout optimization technique [33] to construct a complete connected graph for all variables. In their system, only nodes were colored based on hierarchical clustering.

A recent example for attribute space visualization is the work by Turkay et al. [29], which supports visual exploration in both data space and dimension space. Their system, however, focuses on other statistical quantities, such as mean and standard deviation, and not correlation.

2.2 Unifying Categorical and Numerical Variables

Realistic datasets often contain a mix of numerical and categorical variables. Although there are well-defined statistical techniques to handle categorical and numerical variables in isolation, mixtures of these have received less attention. A few methods [14][22] employed and extended correspondence analysis to transform either categorical or numerical variables into appropriately spaced and ordered categorical variables. We discuss these methods and their limitations more closely in Section 3. Ma and Hellerstein [16] re-order the categories by inter- and intracluster ordering. However, using an equal distance between adjacent categories does not convey the degree of their similarity. In the statistics literature, a popular approach has been to discretize numerical variables into bins and apply statistical methods for categorical variables on them. The inherent problem with this approach – the loss of detail after binning – has been well-reported [23][30]. On the other hand, methods have also been devised that encode categories as numerical values [6][31] which enables one to apply statistical methods designed for numerical data. Typically, however, these methods do not consider the ordering and the distances between categories, which are important features for correlation analysis. We provide a method that also maps categorical variables to numerical ones but optimizes the distances.

2.3 Data Integration

Our method integrates the data into a tessellation derived from the layout of variables. This is another reason why a matrix view is not a feasible option for our system – such a view would not allow a tight data integration. On first glance the visualizations we produce somewhat resemble the hyperbox [1], but we allow scatterplot tiles of more than two variables. Finally, Claessen and van Wijk [5] describe a system that uses tiles of 2D scatterplots and links their axes by the corresponding parallel coordinates display segments. The tiles in our framework are more general and are directly linked at their shared axes.

3 Theoretical Background

The methods available for correlation analysis can be divided into three major groups based on their target variables: (1) methods applicable only to numerical variables, such as Pearson's correlation coefficient; (2) methods applicable only to categorical variables, such as Cramér's V; and (3) methods applicable to computing correlations between numerical and categorical variables, such as the t-test, ANOVA, and MANOVA.

Pearson's correlation coefficient [6] is one of the most popular measures for defining linear relationships between two variables:

$$r(x,y) = \frac{\sum_{i=1}^{N}(x_i - \mu(x))(y_i - \mu(y))}{\sqrt{\sum_{i=1}^{N}(x_i - \mu(x))^2}\sqrt{\sum_{i=1}^{N}(y_i - \mu(y))^2}} \qquad (1)$$

where x and y are two vectors of the same size, $\mu(x)$ and $\mu(y)$ are their respective means, and N is the number of data points. The correlation, r, ranges from -1 to $+1$. The closer r is to -1 or $+1$, the more closely the two variables are linearly related, whereas r close to 0 means that there is no linear relationship between the two variables.

Cramér's V [6] is computed from the χ^2 statistic and can be applied to two categorical variables of any type:

$$v = \sqrt{\frac{\chi^2}{N(k-1)}} \qquad (2)$$

Here, χ^2 is derived from Pearson's chi-squared test and k is the number of rows or columns, whichever is smaller. The metric v ranges from 0 to 1. The closer v is to 1, the more association the two variables have, while a value of v close to 0 means no association between them; v equals 1 only when the two variables are identical. Similar to Pearson's correlation coefficient, Cramér's V is a symmetrical measure – it does not matter which variable is placed in the columns and which in the rows. Also, the order of categories in the rows/columns does not matter. This makes it an appropriate general-purpose measure.

In comparison, the results from the t-test, ANOVA, and MANOVA, which can handle both numerical and categorical variables, are not normalized. This means that they will have different values under different conditions. Thus, to determine if a relationship is strong or not, one must consult specific significance tables. This makes these measures awkward to integrate into an interactive application such as ours. Hence, it is best to first transform a pair of mixed variables into a homogenous pair, either both categorical or both numerical, and then apply Cramér's V for categorical variable pairs and Pearson's equation for numerical variable pairs. Next, we describe alternatives to resolve these mixed variable pairs.

3.1 Dealing with Mixed Variable Pairs

When dealing with two categorical variables it is often important to define a proper ordering and spacing of the individual categories (levels) of each variable. The aim is to order and space the levels of one categorical variable with respect to the other, and the essential task here is to gauge the distribution similarity of the levels of the first

variable with those of the second variable. Levels with similar such distributions are then spaced closer together and others are spaced further apart. This is a global optimization problem and is commonly solved with Correspondence Analysis (CA). Starting from a contingency table, CA computes the set of independent components—similar to PCA for continuous variables. Then the projective coordinates of the first independent dimension give the transformed numerical values. Rosario et al. [22] devised a method that used CA in the context of parallel coordinates. They then extended the scheme to Multiple Correspondence Analysis (MCA) in which the operations are performed with respect to all categorical variables.

For mixed variable pairs, Johansson et al. [14] proposed to first transform all numerical variables into categorical ones and then use the techniques prescribed for categorical values from thereon. They offered two transformation techniques—an interactive approach tied to a parallel coordinate visual interface and an automated one based on *k*-means clustering. However, transforming numerical to categorical variables always results in some amount of discretization, unless the objective number of bins equals the set of unique numerical values. But as the number of bins increases, the computation of any subsequent analysis becomes exceedingly expensive. This makes interactive application of this approach difficult.

Given the disadvantages associated with the typical numerical-to-categorical transformation approach, we chose to devise a scheme that goes the other direction – from categorical to numerical. This avoids the problems with binning and enables the use of Pearson's correlation exclusively. It also allows us to find, for a given categorical-numerical variable pair, the spacing and order of the categorical variable's levels that maximize correlation.

Lastly, we need to define the scope of our transformations. Stevens [27] distinguishes four scales of measurement – nominal, ordinal, interval, and ratio. Nominal variables are the least restrictive. They have no intrinsic ordering in their levels, no fixed interval between them, and no natural zero point. Examples are gender, occupation, or color attributes. Ordinal variables have an intrinsic ordering, such as rankings. Interval variables have an intrinsic ordering and fixed intervals. Finally, ratios also have a natural zero point. There are varied opinions which of these four scales fall into the set of categorical variables. The visualization literature [14][16][22] generally considers only nominal variables which allows a reordering of the levels.

3.2 Regression Model for Categorical Variables

As mentioned, the coefficient of determination r^2 is the square of the correlation coefficient, r, and as such ranges only between [0, 1]. It indicates how well the data points fit a line (if it is a linear model). The objective function for least squares regression is the *residual sum of squares* (*RSS*) which is the data variance unexplained by the regression model. The goal is to minimize *RSS* which maximizes r^2 (and therefore r) since r^2=1-*RSS*/*TSS* [1]. *TSS* is the *total sum of squares*—the sum of squared deviations of the dependent variable values from their mean. We note that since we maximize r^2 the correlation factors that result will always be maximally positive. This, however, is no contradiction since we can always reverse the transformed data axis to reverse the sign of the correlation factor as well.

Regression deals with categorical variables via the introduction of *dummy variables*. There is one such variable for each categorical level minus 1. Let us assume we have an independent, 3-level categorical variable and a numerical dependent variable. This results in the following regression model:

$$Y_i = \beta_0 + \beta_1 I_{1i} + \beta_2 I_{2i} + e_i \quad (3)$$

where $0 \le i \le N-1$, Y is the dependent variable, $I_{1,2}$ are indicator variables (value 0 or 1), $\beta_{0,1,2}$ are the coefficients returned by the regression, and e is the normally distributed error. The indicator variables are only set to 1 when the data point indexed by i is at the corresponding categorical level. The baseline $Y_i = \beta_0 + e_i$ at which neither I_1 nor I_2 is set to 1 is when the third categorical level is active. This model is solved via least squares optimization as usual. Since there is only one dependent variable, this model is called a *univariate multiple regression model*. It can be written in matrix form as $y=X \cdot b$ where y is the $N \times 1$ vector of N observations, b is the $M \times 1$ vector of coefficients with M categorical levels, and X is the $N \times M$ independent variable matrix with indicator values I.

The task we are faced with in our specific problem is to determine the spacing and ordering of the levels of a categorical variable with respect to a numerical variable. Conceptually, this can be accomplished by solving a regression model in which the categorical variable is the independent variable—one dummy variable per level less baseline—and the numerical variable is the dependent variable. The coefficients returned by the least squares optimization then determine the desired order and spacing of the corresponding categorical levels.

3.3 Multivariate Regression Model

In our application a categorical variable may participate in more than one pairing with a numerical variable. Using the arguments in Section 3.2 this is equivalent to having more than one dependent variable. It extends the univariate multiple regression model to a *multivariate multiple regression model*. In matrix form such a model is written as $Y=X \cdot B$ where Y is the $N \times P$ dependent variable matrix with P paired numerical variables, B is the $M \times P$ coefficient matrix, and X is the $N \times M$ independent variable matrix as before. In multivariate multiple regression, each column of B is solved independently and hence there are different dummy variable coefficients for each of the P numerical variable pairings this categorical variable has. The implication for our transformation scheme is that the transformed categorical variable will potentially have different level orders and spacings, one for each of its P numerical variable parings, maximizing the correlation.

[1] $RSS=\sum(y_i - \hat{y})$ and $TSS=\sum(y_i - \bar{y})$ where $\hat{y}$ is the predicted value of y given x, using the regression equation (3), y_i is the actual observed value of y, and $\bar{y}$ is the mean of y.

4 TRANSFORMING THE CATEGORICAL VARIABLES

Dataset	Variable pair categorical/numerical	Corr (Rand)	Corr (Opt)
Auto	make/length	0.115	0.831
	make/price	0.152	0.8871
	make/MPG	0.051	0.712
	make/HP	0.049	0.690
Car	origin/HP	0.034	0.573
	origin/weight	0.112	0.620
	origin/MPG	0.145	0.579
	origin/acceleration	0.054	0.322

(a)

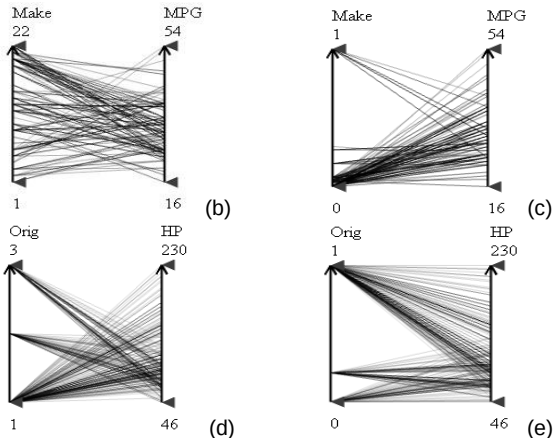


Fig. 2: (a): Correlation coefficients obtained by randomly assigning an integer value 1 through *M* to each of the *M* categories (Rand) and by using the spacing and ordering computed by our optimization (Opt): The correlations achieved by optimization are significantly higher, in many cases by an order of magnitude, for both datasets we tested: Auto and Car. (b), (c) and (d), (e) are two pairs of the parallel coordinate tiles where the visual improvement after the transformation can be informally clearly observed.

Although a regression model serves as a good theoretical background we do not need to solve one to determine the transformation. Instead, we can simply minimize *RSS*. Suppose we are given a dataset Ω with two variables: one categorical variable v_c and one numerical variable v_n. Let us assume there are N data points and M levels in v_c. Let M^i be the total number of data points that fall into categorical level $v_c(i)$ and let $v_n^i(j)$ represent the j^{th} numerical data point that falls into category level $v_c(i)$. The goal is to transform each categorical level $v_c(i)$ in v_c to numerical values $v_c'(i)$ that maximize r. As discussed in Section 3.2, maximizing r^2 (and therefore r) is equivalent to minimizing *RSS* to yield the *RSS* of the transformation, *RSS'*:

$$RSS' = \sum_{i=1}^{M} \sum_{j=1}^{n=M^i} (v_n^i(j) - v_c'(i))^2 \quad (4)$$

Letting $\mu(v_n^i)$ be the mean of all numerical data points that fall into categorical level $v_c(i)$ allows a sequence of manipulation of Eq. (4) which are outlined in Appendix 1. We then arrive at the following expression that needs to be minimized:

$$\sum_{i=1}^{M} \sum_{j=1}^{n=M^i} (\mu(v_n^i) - v_c'(i))^2 \quad (5)$$

Minimization occurs when:

$$v_c'(i) = \mu(v_n^i) \quad (6)$$

Hence, *RSS'* is minimized for a transformation at which each categorical level $v_c'(i)$ is the mean of the corresponding numerical values falling into it, $\mu(v_n^i)$. This scheme is not only elegant but also computationally very efficient since it only requires the calculation of a set of means.

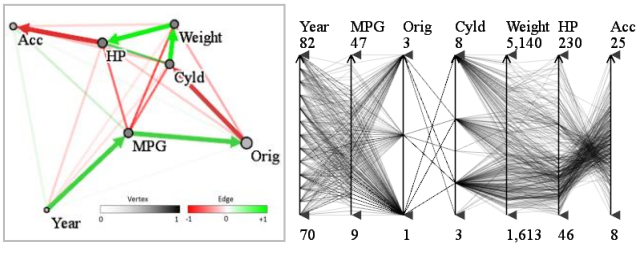
Lastly, the extension of this method to parings of a categorical variable to multiple numerical variables is straightforward since each pairing can be treated sequentially and in isolation, as was shown in Section 3.3.

4.1 First Transformation Results

Fig. 2a shows how this optimization performs for the Auto MPG (car) dataset [34] (N=398) and the automobile dataset [35] (N=205). In the table, the second column gives the variable pairs, and the third column shows the outcome for a random value assignment for each level. The fourth column shows the (greatly improved) correlations obtained with our optimization method. Fig. 2b, 2c and 2d, 2e show two pairs of parallel coordinate tiles before and after the transformation, respectively. We can informally observe that after the transformation, (1) categories (levels) that behave similarly are put close to each other; and (2) the correlation is more visible in the parallel coordinate plots.

5 THE CORRELATION MAP

Our visual exploration framework employs two coordinated displays for dual-space analytics – the correlation map display and an adjunct parallel coordinate (PC) display. Both are shown in Fig. 3. The correlation map in Fig.



(a) (b)

Fig. 3: Our visual exploration framework employs two coordinated displays for dual-space analytics: (a) the correlation map display, and (b) an adjunct parallel coordinates (PC) plot. This example uses the car dataset. The route in the correlation map indicates the axis order in the parallel coordinate display. We observe that cylinders (Cyld) and weight are close in the correlation map and positively correlated (green-colored edge). The PC plot confirms this – the lines do not overlap much. Conversely, we observe an approximate bowtie shape in the horsepower (HP) – acceleration tile (Acc is proportional to time, not 1/time) and indeed these variables are also appropriately close in the correlation map with a red-colored edge for negative correlation. Finally, Year and MPG are more distant in the correlation map which is confirmed by the reduced line structure in the PC plot.

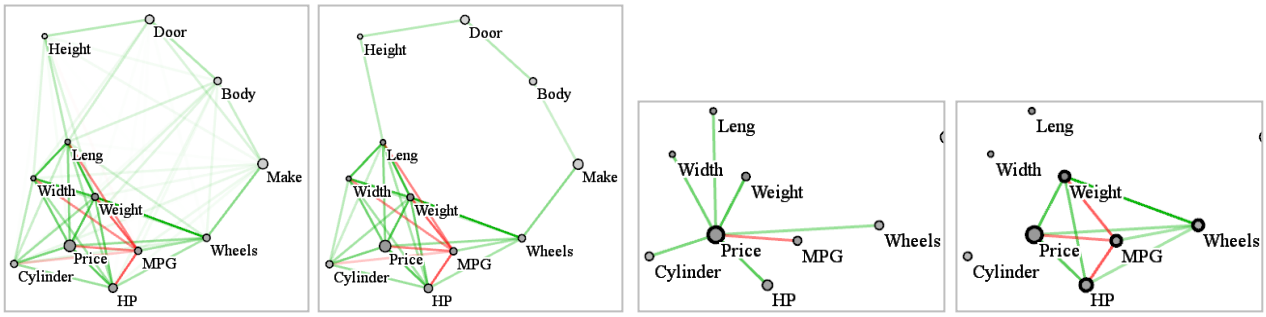


Fig. 4. (a) Correlation map for the automobile dataset. (b) The edge correlation filter δ_e is set to 0.5 to remove low correlation edges. To obtain (c) and (d), the user first filters out *Height*, *Body*, *Door* and *Make* due to small accumulated correlations. (c) The *vertex-browsing* correlation map with focus only on *Price*. All edges incident on *Price* are highlighted. We quickly see that the type of drive-wheels (Wheels – 4×4, fwd, rwd) and car length (Length) have a much smaller effect on Price than horsepower (HP) and MPG, which is negatively correlated since, as we learn, more expensive cars also cost more gas. (d) The *edge-browsing* correlation map after selecting a group of variables (shown with thicker outlines). Only correlations (edges) within the group are shown to help users to focus on this local set of variables. This view helps users appreciate the 'eco-system' of Weight, Wheels, Price, and HP centered about MPG. Via this graph, users can quickly learn about the relationships and tradeoffs that exist among the car properties closely related to MPG. It can be seen in one view that high MPG requires low weight and low HP cars and that the type of wheel drive (Wheels: fwd, rwd, 4×4) also matters. We further confirm that cars with these properties are fortunately inexpensive (which is also suggested by the negative correlation of MPG with Price).

3a provides an overview of all dimensions in terms of their pairwise correlations in variable space, whereas the PC display in Fig. 3b shows the raw data with sequentially ordered dimensions in data space and serves as an additional manipulation interface for the correlation map. Our work presented in [33] describes how the correlation map can be transformed into a network and be used to control the axes ordering in the PC display. The directed path of thick lines in Fig. 3b denotes an optimal (but user-configurable) route found by running a Traveling Salesman solver over the network and ordering the PC axes accordingly. This path ensures that neighboring PC axes exhibit a high degree of correlation, which promotes the discovery of salient relationships in the PC display. The caption of Fig. 3 has a narration of some of the findings that can be made for the car dataset [34] using this display.

5.1 Interacting with Correlations

In the correlation map (Fig. 3a), vertices correspond to variables. The vertices are laid out via a mass-spring model [12] in which spring rest length is a function of the absolute value of the pairwise correlation strength among variables. The mass-spring model seeks to produce a layout in which the spatial distance among each pair of vertices is equivalent to their spring rest length. The vertex size, on the other hand, encodes a variable's standard deviation (variance). We chose this coding because variance is an important factor in correlation analysis. Mapping correlation strength to spatial proximity and variance to vertex size allows users to quantitatively assess and compare these statistical properties and do so in a combined and holistic map layout display.

5.1.1 Visual Edge and Vertex Enhancement

In correlation analysis, if a variable is highly correlated with other variables, it often plays an important role in the analysis process. We use the accumulated correlations from all other variables to compute the significance factor, R_i for a specific variable v_i:

$$R_i = \frac{\sum_{j=1}^{D} |\, correlation(v_i, v_j)\,|}{D-1},\ j \neq i \qquad (7)$$

Here, D is the number of variables and the denominator normalizes R_i. A value of $R_i = 0$ means that variable v_i has no linear relationship with any other variables; while $R_i = 1$ means that v_i has a strong linear correlation with all other variables. We use opacity to visually encode R_i.

Conversely, the edge significance is weighted by the correlation between the two dimensions linked through the edge. The correlation is encoded by color and opacity. Green encodes positive correlation while red encodes negative correlation. Edge opacity is determined by the strength of correlation.

5.1.2 Focus + Context Browsing

In the correlation map, visual clutter can arise when the number of dimensions grows large (Fig. 4a). Since the correlation map is a complete graph, the high dimensionality makes it difficult to follow the edges [8]. We provide several options to help reduce the clutter, following the information seeking mantra: overview first, zoom and filter, then details-on-demand [25].

Specifically, we provide two correlation filtering operators for vertices, and one such filter for edges. The filtering operators for vertices are used to control two objective parameters – accumulated correlation, R, and standard deviation, σ – which are controlled by two sliders defining thresholds δ_R and δ_σ. Only vertices with $R_i > \delta_R$ and $\sigma_i > \delta_\sigma$ are deemed significant and shown in the map. The filtering operator for edges allows users to define a threshold, δ_e. For correlation strengths (absolute value) smaller than δ_e, the corresponding edges will be filtered out from the correlation map, as shown in Fig. 4b. These filtering operators help users guide their focus on highly correlated variables only (those with r>0.8) but dismiss weak correlations (those with r<0.5).

Detail-on-demand is supported by two interactive different browsing modes: *vertex-browsing* mode and *edge-browsing* mode. In *vertex-browsing* mode (Fig. 4c), hover-

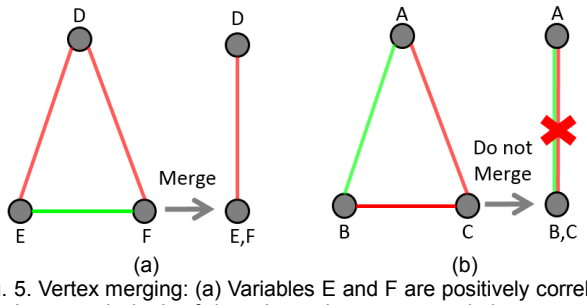


Fig. 5. Vertex merging: (a) Variables E and F are positively correlated. As a result, both of them have the same correlation type with variable D. Then, when merging E and F, the merged vertex retains the same type of correlation as before: *edge*(D, EF) vs. *edge*(D, E) and *edge*(D, F). (b) Variables B and C are negatively correlated. Therefore, they have different correlations with variable A. If we merged B and C, the edge from BC to A would be inconsistent with those before the merging: *edge*(A, BC) vs. *edge*(A, B) and *edge*(A, C). Hence, we do not merge negatively correlated variables.

ing over a vertex highlights all of its adjacent edges, and clicking on a vertex keeps the adjacent edges highlighted. Conversely, in *edge-browsing* mode (Fig. 4d), the user can select a group of interesting vertices; only edges with both adjacent vertices inside the group will be highlighted. In both modes, the selected vertices are rendered with thicker outlines to distinguish them from others. These interactions can help users to interactively explore the correlation space for interesting discoveries. The reader is directed to the caption of Fig. 4 to see illustrative use cases for both of these modes, and their findings.

We end by stating that after the mass-spring model layout, the relative locations of vertices alone can already give indications of the correlation strength information among variables. Thus, users can set $\delta_e = 1$ to turn off all edges without losing much correlation strength information. By default, however, all thresholds are set to 0, which shows all edges without filtering.

5.1.3 *Multi-Scale Zooming*

To support the visualization of datasets with many variables in a limited screen space, we refined the multi-scale zooming interface described in [33]. This technique uses the (distance-mapped) correlation strength between vertices to decide whether variables should be merged or not. In our current work we propose a more accurate criterion to control the merging (Fig. 5), and we also add a new rule for selecting the variable to be used for representing the merged variables. Both of these contributions are discussed in the following paragraphs.

When merging variables, one needs to be careful not to abstract correlation information in inconsistent ways. Note that two highly positively correlated variables behave similarly in terms of correlations with other variables (Fig. 5a), while two negatively correlated variables behave differently (Fig. 5b). When considering merging close variables, we merge only those that have positive correlations. Thus, after the merging, the edges adjacent to the merged variable remain consistent with the original edges. With this merging criterion, as users zoom out of the display, nearby variables with positive correlations merge into one, and as users zoom back in, the merged variables split into the original variables. In our implementation, users can also manually control whether to merge or collapse a representative vertex via mouse-clicks.

When considering a representative variable for a set of merged variables, we choose the one with the largest accumulated correlation (R_i). This is justified because R_i not only takes correlation into account, but it is also a data-

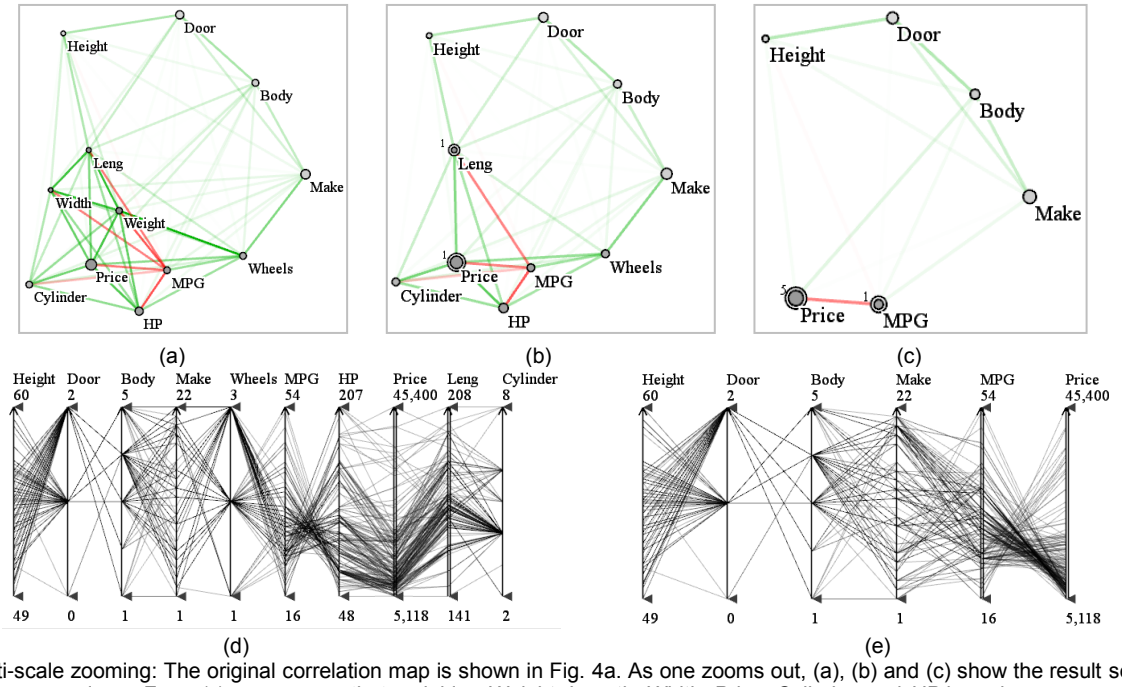


Fig. 6. Multi-scale zooming: The original correlation map is shown in Fig. 4a. As one zooms out, (a), (b) and (c) show the result sequences of correlation map views. From (c) we can see that variables *Weight*, *Length*, *Width*, *Price*, *Cylinder*, and *HP* have been merged into one (*Price*) because all are positively correlated. Although *MPG* is close to them, it has a negative correlation with the others, so it is not merged. But *MPG* and *Drive Wheel* have positive correlation, so they also merge into one (*MPG*). The number of variables packed into the representative variable is given by the small number in the upper left corner of representative vertex. Panels (d) and (e) show the corresponding PCP displays of (b) and (c), respectively. The representative dimensions are denoted by double-line axes.

centric measurement, encoding the variable's significance with regards to other variables. Moreover, the variable with a larger accumulated correlation tends to better indicate or predict the other variables. Another option would be to use factor analysis or PCA to extract the main factor or component as the representative variable. We did not implement this since it is not straightforward to understand what a factor or component actually means, making it hard to interpret relationships from them.

Fig. 6 shows a simple example for how the multi-scale zooming works, again for the car dataset. Zooming activities in the correlation map are also reflected in the adjunct PC display, whose complexity is reduced as well.

5.1.4 Exploring Correlation Sensitivity

Correlation strength can often be improved by constraining a variable's value range. This will limit the applicability of the derived relationships to this value range, but such limits are commonplace in targeted marketing and elsewhere. We shall illustrate this point by ways of an extreme example. Let us assume we have a variable pair x and y, where for half the samples the pairs have the same sign, say $y=x$, and for the other half the pairs have the opposite sign say $y=-x$. Then for the first half $r=1$ and for the second half $r=-1$. This would mean that the total $r=0$ and that the correlation map would display the two variables far apart. However, this is not really a complete analysis. Rather, one should separate the two sub-ranges (if this is meaningful to the task at hand) and look at them in isolation. But recognizing such a situation can be challenging, since it is usually not as clear cut as in this extreme example. The adjunct parallel coordinate display is a good instrument for this – we can isolate the sub ranges via interactive bracketing and view the effects this has in the correlation map. Our system provides such a facility.

Fig. 7 shows a simple example examining the relationship of price and sales for a fictitious product. Fig. 7a shows that there is no correlation when the full price range is considered. However, bracketing on the lower price range yields a positive correlation of price and sales (Fig. 7b), while bracketing on the higher price range yields a negative correlation (Fig. 7c).

5.2 Integrating Data—the Subspace Scatterplot

While the adjunct PC display provides access to the raw data, it requires users to transition their eyes to a different screen area. Further, due to PC's sequential axis ordering, multivariate data relationships are difficult to detect across more than three dimensions. Multivariate scatterplots can overcome these problems (see [18] for example). In the following section we describe our attempt to integrate multivariate scatterplots into the correlation map.

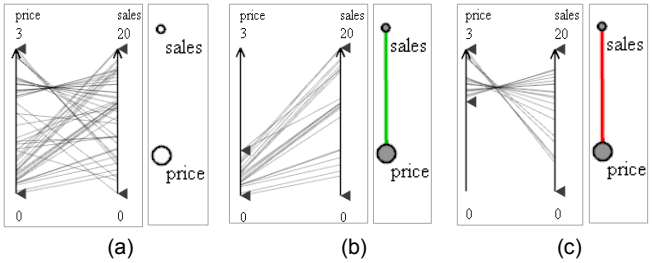


Fig.7: The influence of bracketing on correlation for a fictitious product's price and sales. (a) Full price range, no correlation; (b) bracketed to lower price range, positive correlation; (c) bracketed to higher price range, negative correlation.

5.2.1 Tessellating the Correlation Map

As a first step, we need to create bounded areas into which we can project the data. We accomplish this by tessellating the scattered point set formed by the correlation map's vertices. We use the following strategy:

1. Triangulate the domain using Delaunay triangulation. Delaunay triangulations maximize the minimum angle of all triangle angles in the triangulation. It tends to avoid skinny triangles which are less capable of showing the details of the corresponding subspace. Furthermore, the triangulation will generate a planar graph that avoids edge crossings.
2. Sort all the edges in ascending order by edge length, which is equivalent to sorting the correlation coefficients in descending order.
3. Optionally, for edges with length less than some threshold, (i.e., a correlation greater than some value), if removing the edge will not cause concave polygons, remove it. Concave polygons are not suitable since our data mapping method requires convex primitives.
4. Create the scatterplot for each of these subspaces using the method outlined in Section 5.2.2.

The third operation will yield polygons with more than three vertices and can be used to visualize data distributions due to higher-order subspaces. Since the variables used to create the polygon are close, the dimensions in these subspaces are sufficiently correlated to give rise to meaningful data configurations in the projections.

5.2.2 Generating the Subspace Scatterplots

After the tessellation, the map is divided into a mesh of polygons, each corresponding with a data domain subspace. The next step is to project the subspace data into their associated polygons, generating the subspace scatterplots. For this we require a method that can forward-project a high-dimensional data point p into the geometry of a concave polygon P defined by S vertices q_i ($0 \le i \le S-1$), where S is the dimensionality of p's subspace. The projection of p into P's 2D domain yields a point p' and is a function of the spatial coordinates of the q_i that represent the variables spanning the subspace. In other words, if p's only non-zero coordinate were in variable i, it would map directly to vertex q_i. For all other constellations, the following weighted mapping is utilized, where the weights are the attribute values normalized by the sum of values for all of subspace attributes of p:

$$p' = \sum_{i=0}^{S-1} w_i q_i \qquad w_i = p(i) \Big/ \sum_{j=0}^{S-1} p(j) \tag{8}$$

where the $p(j)$, $0 \le j \le S-1$, (likewise $p(i)$), are the coordinates of p in its high-dimensional subspace and the q_i are the spatial coordinates of the subspace polygon's vertices in the correlation map. This mapping is adapted from the method of generalized barycentric coordinates [17]. The coordinates of p are measured with respect to the subspace origin which in our case is the vertex of the

subspace (hyper) bounding box that has the minimum value in all subspace dimensions. We note that while the relationships among the projected points are not exactly correlations or cosine similarities, they share some properties of these metrics, such as the insensitivity to scaling in high-dimensional space.

Finally, the color of a point is determined by its cluster membership. The mapped points are organized into pixel-bins in the subspace polygons. We record the maximum/minimum extent of the *S*-dimensional bounding box of each point cluster, use it to determine density, and indicate denser bins by higher intensity. There might be cases in which some regions have very high density while most other regions have low densities. The low density points will then become difficult to see after intensity normalization. We provide a slider bar that allows users to control the degree of transparency of the scatterplot. If the value is 0, all points will have the maximum intensity, that is, there will be no transparency at all.

5.2.3 *Reading the Subspace Scatterplots*

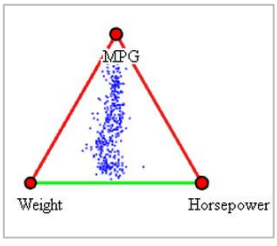


Fig. 8: Subspace scatterplot.

The subspace scatterplot generalizes RadViz [11] from a circle to a generalized polygon. Similar to Radviz, the location of a projected point indicates how much it gravitates towards a particular attribute (or set of attributes). This allows the assessment of biases, trends, and trade-offs. For example, in Fig. 8, we observe a positive correlation between *Weight* and *Horsepower* and a negative correlation between *MPG* and both *Weight* and *Horsepower*. In the corresponding scatterplot we observe that all cars map to a long cluster centered between *Weight* and *Horsepower* and reaching towards *MPG*. This effectively visualizes the trade-off that exists between weight and horsepower—there are no light cars with high horsepower and vice versa – and it also shows that high MPG requires one to lower both weight and horsepower, but that this trade-off function is smooth and continuous.

6 Applications

We have used the following three datasets to demonstrate applications that highlight the features of our framework.

University dataset: This dataset consists of 50 colleges with 14 attributes: *academics, athletics, campus housing, night life, safety, transportation, weather, dining, PhD/faculty ratio, population, household income, USNews score, tuition,* and *location.* The dataset is an amalgamation of data obtained from two different sources: the College Prowler website [36] and US News & World Report [37]. The former ranks each school across the 20 most relevant campus life attributes. We took the top 50 colleges from US News and three attributes *USNews score, tuition,* and *location*. All the other attributes are from from College Prowler. In this dataset, *location* is a categorical variable in terms of city, the others are numerical ones.

Sales campaign dataset: It was obtained from a business intelligence company and has data quantifying parameters in sales and marketing. There are 900 data points (one per salesperson) and 10 attributes: *%Completed, #leads, leads won, #opportunity, pipeline revenue, expected ROI (Return on Investment), actual cost, cost/wonLead, planned revenue,* and *planned ROI*. This is a synthetic dataset, but it was built based on actual models that realistically describe sales behavior. All variables are numerical.

File compression dataset: It captures parameters from computer and file systems and consists of 864 data points and 10 attributes: *file type, compression algorithm, compression level, CPU frequency level, compression ratio, energy, performance, time, CPU temperature,* and *current*. The dataset was captured via experiments that measured different parameters when conducting file compressions. Variable *file type* (all zeros, text, binary, random), *compression algorithm* (gzip, bzip2, lzop, or none), *compression level* (1 to 9), and *CPU dynamic voltage and frequency level* (8 levels) are the inputs, and the others are the measured outputs. *Energy* is measured in Watt-hours. *Performance* is measured as the number of files compressed per second. *File type* and *compression algorithm* are categorical variables.

6.1 Correlation Analysis: The University Dataset

We first demonstrate the basic concepts of our framework with the university dataset. Fig. 9a shows the correlation

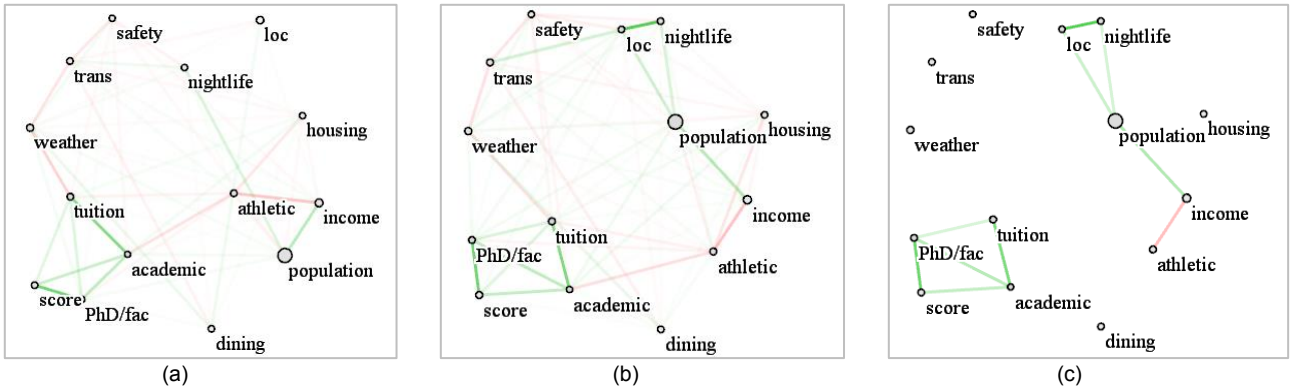


Fig. 9. Correlation map for the University dataset: (a) Original correlation map generated by assigning random numerical values to variable *location (loc)* – no significant correlations seem to exist. (b) Correlation map after transforming *loc* with our algorithm – now *loc*'s positive correlation with *nightlife*, *population* and *transportation* are apparent. (c) After applying the edge correlation filter (setting δ_e=0.3) – the data divide into two fairly independent clusters – one (lower left) dealing with academic aspects, the other (upper right) with student life.

map, in which the attribute *location* was randomly assigned to numerical values to compute correlations. We see that there is no strong correlation with any other variable. Next we apply our transformation algorithm and the corresponding correlation map is shown in Fig. 9b. Now we observe that *location* is in fact quite strongly correlated with a number of variables nearby, such as *night life*, *safety*, *transportation* and *population*. The last attribute is particularly interesting in that its large vertex size indicates that the dataset contains a large variety of university settings—urban, suburban, and rural.

From Fig. 9b we observe that the majority of correlations are not overly strong, as is apparent from the mildly saturated edges and vertices. So in order to isolate the more significant correlations we raise the edge correlation threshold to δ_e=0.3. The resulting map is shown in Fig. 9c where we observe two fairly independent clusters—one dealing with academic aspects, the other with student life. This reveals that these two aspects of the college experience tend to be largely independent in general.

Analyzing the map

We observe that the correlations within either of these clusters are mostly positive (indicated by the green edges). In the 'academic' cluster at the bottom left of Fig. 9c all variables (*US News Score, PhD/Faculty ratio, Tuition,* and *Academics*) are positively correlated with one another. Hence, when one variable increases, all others will increase too, and vice versa. This observation is consistent with our knowledge that highly ranked universities (high *US News Scores*) usually have better *academics* and higher *tuition*. Yet, because students are more willing to go there, the *PhD/faculty ratio* is higher.

On the other hand, in the 'student life' cluster on the right of Fig. 9c we observe that *athletics* is negatively correlated with *income*, whereas *income* is positively correlated with *population*. A possible explanation for this is that the universities with good athletics are usually located in rural areas, which are less densely populated, and the income in these areas is relatively low compared to other more populated areas (e.g., New York City). We also find that *night life* has a high positive correlation with *location* and *population*, which is also justifiable.

Finally, in Fig. 9b we observe that one of the stronger connections between the academic and student life clusters is the (negative) correlation of academics and athletics. Indeed, athletics, like football or basketball, are often public hallmarks of a university. The negative correlation clearly shows that academically highly ranked schools, which are most often private, typically do not have nationally visible athletic teams, with some exceptions.

Overall impact

Many more conclusions can be drawn from this single visualization. Therefore we believe that these maps can be helpful for students to select universities, as well as for university executives to plan policies and campaigns.

6.2 Case Study: The Sales Campaign Dataset

We use the sales campaign dataset to show how our framework can help business executives in making marketing decisions. Let us first review some basics. The typ-

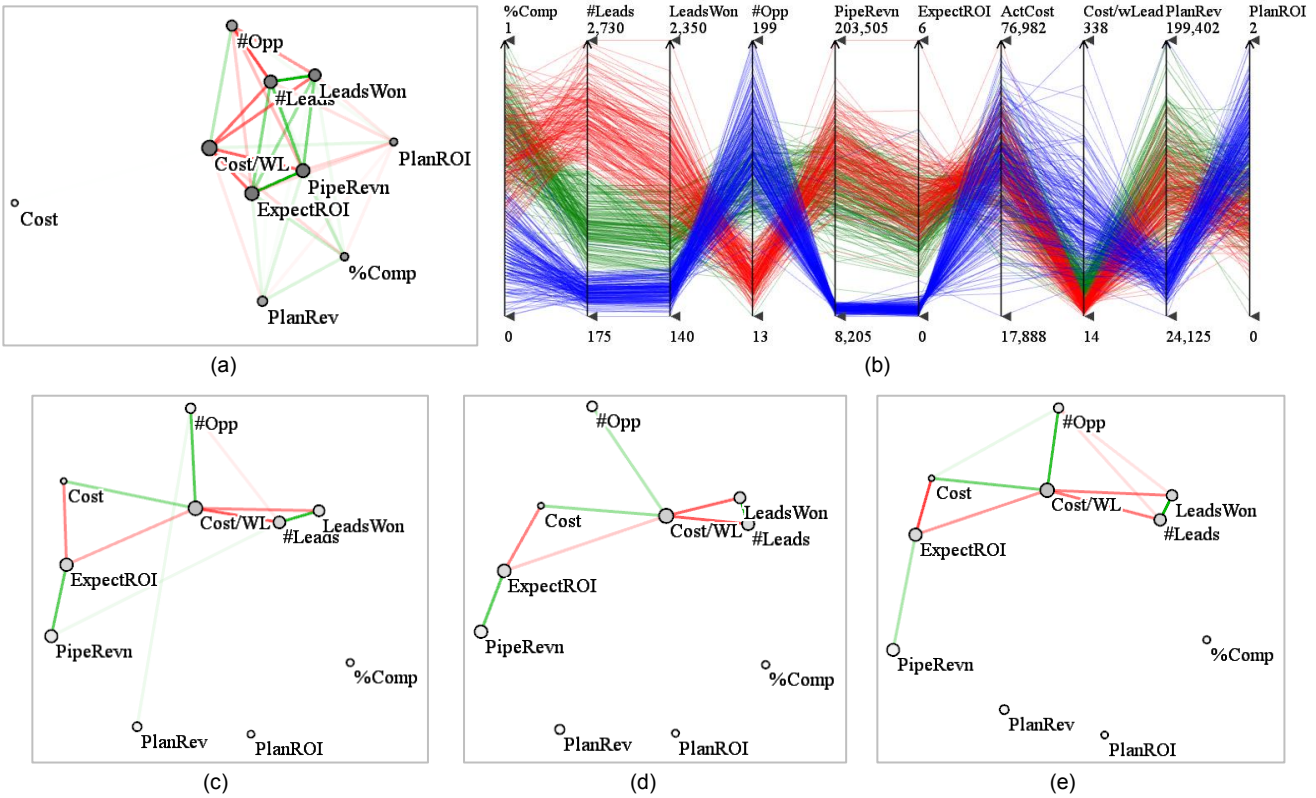


Fig. 10. Business strategizing with the sales campaign dataset (a) Aggregate correlation map for all three sales teams. (b) Parallel coordinate plot for the three sales teams (clusters), colored red, green, and blue. (c)-(e) Correlation maps for the red, green, and blue teams, respectively. All edges with correlation strength less than 0.2 have been filtered out to extract the main structure of the map.

ical corporate sales pipeline begins with a *lead generator* whose job is to produce a number of prospective customers that have some level of probability that a salesperson will actually close a deal with them. Upon a positive response, these *leads* become *won leads (qualified lead)* and receive an increased sales pitch at a *cost per won lead*. If this pitch wins further positive response, then these won leads become *opportunities*, which might be potential customers in the future. In practice there are many more levels, but this may serve as a sufficient model here. Of course, cost is involved in each step of the pipeline, and the eventual *pipeline revenue* is the ultimate important factor. In [33] we outlined an example that used the correlation map's network interface to determine an ordering of the parallel coordinate axes by which the best sales strategy among three sales teams could be discovered. For this paper, we place more focus on the correlation map itself.

6.2.1 Business Strategizing

As a practical scenario, let us imagine a meeting of company executives who would like to make sales policies for the next year based on their three sales teams' behaviors of this year. John from the marketing department always wants more opportunities. By looking at the correlation map of the three teams (Fig. 10a) he states that since *cost* does not have strong correlations with other variables, the company can make any strategies for other variables, and it will not influence the actual cost. So, he proposes that the company should improve efforts to create more *opportunities* for the next year without considering the money issue. Based on the correlation map, such efforts could be reducing the number of *leads* and *won leads*, thus increasing the *cost per won lead*.

Consulting the Parallel Coordinate display

Emily, from the financial department, believes that there must be something wrong with this statement since *cost* should play an important role in the sales pipeline. By looking at the data space, the PCP plot, which is shown in Fig. 10b, she notices that these three sales teams behave quite differently. It is likely a mistake if they consider the three teams together. Hence, she suggests that they plot the correlation maps for the three teams separately. The results are shown in Fig. 10c, d, and e, for the red, green, and blue teams, respectively. It is interesting to note that the three teams have quite similar correlation patterns, which is consistent with her expertise that there must be some marketing model that guides the sales behaviors and the model should involve *cost* in it. From the plots, one can see that there are 7 variables involved in the pattern: *opportunities, cost, cost per won lead, lead, lead won, expected ROI,* and *pipeline revenue*; other variables are not as closely related. As a result, these 7 variables should be focused on as references to make decisions.

Detailed analysis with the correlation map

Based on these observations, Emily claims that the actual influences of increasing the *opportunities* should be: (1) *cost per won lead* will be increased because it is the only one that is related to *opportunities* in the plot, with positive correlation. However, *cost per won lead* is highly correlated with four other variables. As a result, (2) the number of *leads* and *won leads* will be decreased due to the negative correlations; (3) the *cost* will be increased and the *expected ROI* will be decreased. So she proposes to reduce the *cost* for the next year. The corresponding impacts are: (1) the *expected ROI* will be increased due to its negative correlation with *cost*, which is another good factor; (2) the *cost per won lead* will be reduced due to its positive correlation with *cost*; (3) the *opportunities* will also be reduced which is a negative effect. After listening to these two proposals, CEO Tom is about to make final the decision. First, increasing *cost* is not preferred because this year's expense already exceeds the budget. Second, although the number of *opportunities* is reduced, the *expected ROI* will go up. By considering these conditions, Tom decides that the policy should follow Emily's proposal.

6.2.2 Subspace Scatterplot Based Analysis

Manipulating the subspace scatterplots can also reveal many interesting relationships, in situ with other information in the correlation map. Then, by switching to the parallel coordinates these relationships can be examined more quantitatively. Let us now look at a few examples.

Visualizing clusters and priorities

Fig. 11a shows the correlation map of Fig. 10a now augmented with the subspace scatterplots that were automatically generated by our tessellation algorithm. Already at

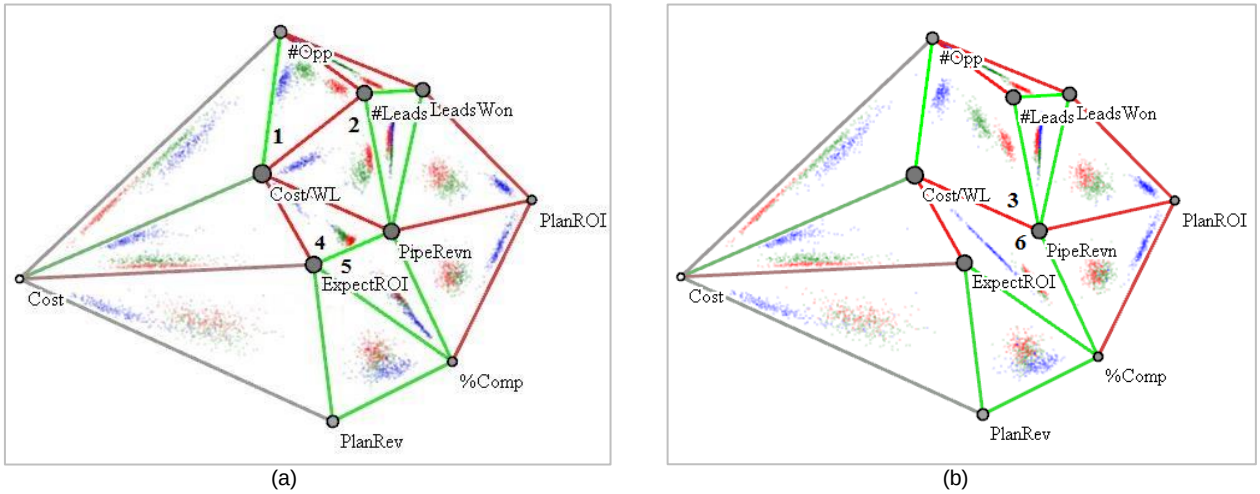


Fig. 11: Scatterplot analysis: (a) default layout, (b) mesh after removing the edge between subspace 1 and 3 as well as 4 and 5.

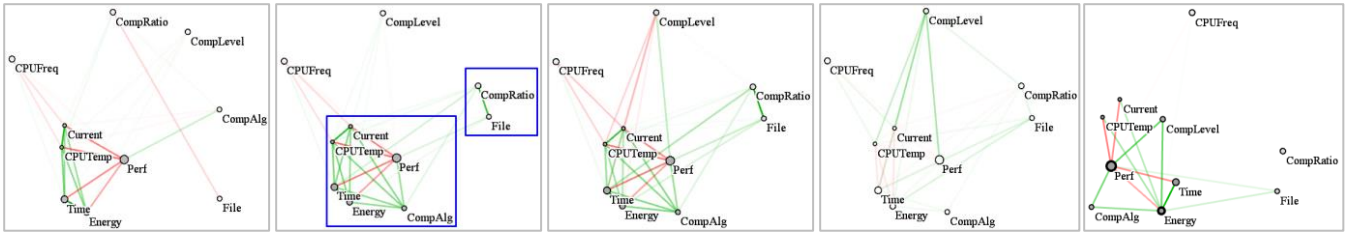


Fig. 12. Controller design for the file compression dataset: (a) Correlation map generated by assigning random numerical values to the categorical variables. (b) Correlation map computed by our new transformation algorithm. (c) Correlation map after selecting only high compression levels. (d) Correlation sensitivity map change from (b) to (c). (e) Correlation map for data with high compression levels only, laid out by the correlations of the data with high compression levels.

first glance it becomes obvious that the blue team is quite different from the green and red teams—the latter two clusters overlap often while the blue cluster is disparate in most subspaces. In subspace 1 we observe that the red team's focus seems to be on generating many leads—but only few converted into opportunities. On the other hand, the blue and green teams seem to have better priorities by focusing on opportunities instead of leads. In subspace 2 we learn that the blue team spends much money per won lead, but this does not seem to translate to high pipeline revenue. In this aspect the red and green teams do better.

We now wish to get a more comprehensive picture about these issues and remove the edge separating these two subspaces. This generates subspace 3 in Fig. 11b. Now we see the three teams well separated and learn that while the blue team does great on winning opportunities, it does poorest among all teams in terms of final pipeline revenue, while incurring more cost (relatively speaking) than the other two teams. Surprisingly, the red team (and also the green team to a lesser extent), despite the relatively few opportunities it creates, has a much better pipeline revenue emphasis than the blue team, possibly because it spends little money on its leads.

Finding appropriate subspace dimensionalities

Our tessellated map effectively "unrolls" the high-dimensional space into the plane as a mesh of subspace scatterplots. But merging some of the plots can bring an even better understanding. In the previous example, we saw that for neither subspace 1 nor subspace 2 all three clusters could be separated at the same time. This indicates an insufficient number of dimensions for the subspaces the three clusters reside, and indeed we saw that by merging the two subspaces the clusters could be well separated. Hence, this configuration of the scatterplot mesh constitutes a better unrolling of the high-dimensional space, accounting for the intrinsic dimensionality of subspace 3. Similar is true for subspaces 4 and 5 in Fig. 11a where we see strong overlaps for some of the red and green cluster. Merging these two subspaces into subspace 6 (Fig. 11b) has a similar effect than with subspace 3—the three clusters are now much better separated.

6.3 Application in Control Theory

Given a complex system (plant), the goal of control theory is to develop techniques for the semi-automatic synthesis of a controller. This requires the existence of models of the plant and there are two types of models: Single-Input-Single-Output (SISO) model and Multiple-Input-Multiple-Output (MIMO) model. A SISO model has only one input parameter and one output parameter while a MIMO model could have multiple input parameters and multiple output parameters. Models are learned from an approximation of the input and output behaviors of the plant. The choice of the input parameters is very important: they should expose the associations with the outputs. However, the multitude of parameters and associations are buried in various configuration scripts. Finding these parameters and associations from the many inputs and measured outputs poses challenges to analysts, engineers, and researchers alike.

For parameter selection, unfortunately, the plants have a large number of compile-time and run-time parameters, which poses challenges to model identification. Take the SISO model, for example. Suppose there are N_I measured inputs and N_O measured outputs; in the worst case, the users need to try every possible combination of the input variables and output variables (which are N_I*N_O possible cases) for model identification. The MIMO model could result in even more possible cases. Thus, it is infeasible to explore all possible model settings. As a result, efficient tools are needed to suggest a manageable subset of controllable parameters. The guidance for parameter selection is that within one model, the input(s) and output(s) should be highly correlated with one another. At the same time they should have low correlation with other parameters that do not belong to the model.

6.3.1 Initial Analysis with the Correlation Map

Fig. 12 narrates how our correlation map can help analysts in the selection of parameters for model discovery. Fig. 12a is generated by randomly assigning categorical variables to equal distance numbers (1, 2,…, M). We observe one dense cluster in the lower left corner, but it is formed solely by numerical variables—all categorical variables (inputs) remain on the map's boundary, indicating only limited correlation with the other variables. This is problematic since it suggests that there is no relationship between the inputs and the outputs which is doubtful. To resolve this problem, the analyst transforms the categorical levels into numerical values. The outcome is shown in Fig. 12b. We now observe two clusters (blue boxes). The cluster in the upper right blue box contains only two variables: *file type* and *compression ratio*. Their correlation is highly positive with almost no correlation with other variables. This suggests that they form a good SISO model.

However, the model of interest to the analyst focuses

on *energy* and *performance*. These variables are part of the other cluster, shown in the lower left blue box. He notices that *compression algorithm* is in the cluster while *compression level* is not. Yet, drawing on domain expertise, the analyst knows that *compression level* should have something to do with the other parameters, such as *performance* or *current*, but he suspects that this might only occur at sufficiently high levels. So he uses the parallel coordinate display (not shown) to interactively select the data points that have higher values in their *compression levels*. The corresponding correlation map is shown in Fig. 12c. The map reveals strong correlations, both positive and negative, suggesting a correlation function and control model only valid within a certain range of compression levels.

6.3.2 Using the Correlation Sensitivity Map

Having discovered that filtering out the lower compression levels adds strength to the overall model, the analyst would like to know which parameters in particular benefit. To provide visual support in this analysis, we apply a different edge encoding scheme in our correlation map—one that emphasizes the sensitivity of relationships with respect to value bracketing. We call this visualization the *correlation sensitivity map* (Fig. 12d). In this map, an edge is colored by the *change* in correlation strength (absolute value) instead of absolute strength. This quantity, $\Delta corr$, is obtained by subtracting the (absolute) unfiltered correlation from the (absolute) filtered one:

$$\Delta corr = |corr_{filtered}| - |corr_{unfiltered}| \quad (8)$$

In this scheme, a green edge indicates that the strength of the correlation between the two corresponding variables has increased after the change, while a red edge indicates that the strength has decreased. From Fig. 12d, the analyst learns after disqualifying low *compression levels*, the correlation of this variable with the *energy-performance* cluster can be made highly significant. This then renders the cluster variables much more predictable, which is very helpful for model identification.

Finally, we note that the correlation sensitivity map still uses the map layout of the unmodified data configuration. To visualize the new correlation relationships, the analyst re-layouts the correlation map based on the new correlation strengths computed from only the data points of high *compression levels*. Fig. 12e shows this new layout with a focus on the two important output parameters: *performance* and *energy* via the *vertex-browsing* mode. The analyst quickly learns that both *compression* parameters—*algorithm* and *level*—are significant to *energy* use and *performance*, indicated by their strong positive correlation to them. This also means that both of them should belong to one model. Hence a MIMO model is required here.

7 Discussion

We showed that correlation analysis is useful for exploring the relationships between pairwise variables and for predicting one variable's behavior based on another. However, a main drawback of correlation analysis is that a relationship between two variables does not imply a causal effect—any two variables could be correlated, but this does not necessarily mean that one is the cause of the other. One example is found in the automobile dataset (Fig. 4d). *Price* has a strong positive correlation with *Weight*. We can only say that given a high price, usually we can predict that the car has high weight, or vice versa. But we cannot say that high price causes high weight, or vice versa. The other drawback of the correlation is that it applies only to variable pairs. Sometimes we need multiple factors to explain one behavior. Just as shown in Fig. 12d, we can see that *performance* and *energy* exhibit a strong negative correlation with each other. Yet, we cannot draw the conclusion that high-performance compression algorithms will consume low energy. To explain this behavior, we need to include other variables—time and task. Given a specific task, high performance algorithms use less time (red edge), and as a result, they consume less energy (red edge). This makes sense since energy is performance integrated over time, and so a causal network path between energy and performance must include time. Nevertheless, we note that the purpose of our framework is not to provide innovations in addressing limitations of correlation analysis, but to provide an efficient tool to help analysts to do interactive correlation analysis and predictions.

8 Conclusion and Future Work

We have presented an interactive framework that enables correlation analysis and visual association mining for high-dimensional data. Our correlation map can serve both as a data exploration environment and as a platform to visually demonstrate, explain, and justify associative relationships that exist in the data. Our framework is quite general and applicable to a wide set of applications. It handles both categorical and numerical variables, scales to large numbers of variables via a multi-scale semantic zooming approach, and allows interactive value bracketing to discover correlations hidden in value intervals. The correlation map also allows users to interactively visualize data relations within the subspaces spanned by correlated variables by projecting the data into a corresponding tessellation of the map.

A present limitation is that correlation can show only pairwise relationship of two single variables, but strong relationships may exist between two sets of variables. For example, while area (=width*length) could be correlated with price, neither width nor length might be. Possible solutions to explore are regression and subspace analysis.

Also, correlation can be affected by outliers, non-linear relationships, heteroskedasticity, and multicollinearity. To gain more statistical robustness, we would like to use techniques for outlier detection and/or removal, and methodologies for detecting and visualizing non-linear relationships and relationships among multiple variables.

Finally, the points projected into our subspace scatterplots do not indicate the vector magnitude in data space. They only indicate relative proportions in the attributes spanning the subspace. We hope to study how visual attributes such as intensity, saturation, opacity, and size (e.g. [21]) can be employed to visualize this strength.

APPENDIX – DERIVATION OF MINIMIZATION

$$RSS' = \sum_{i=1}^{M}\sum_{j=1}^{n=M^i}(v_n^i(j) - v_c'(i))^2$$

$$= \sum_{i=1}^{M}\sum_{j=1}^{n=M^i}(v_n^i(j) - \mu(v_n^i) + \mu(v_n^i) - v_c'(i))^2$$

$$= \sum_{i=1}^{M}\sum_{j=1}^{n=M^i}(v_n^i(j) - \mu(v_n^i))^2 + \sum_{i=1}^{M}(\mu(v_n^i) - v_c'(i))^2$$

$$+ 2\sum_{i=1}^{M}\sum_{j=1}^{n=M^i}(v_n^i(j) - \mu(v_n^i))\sum_{i=1}^{M}(\mu(v_n^i) - v_c'(i))$$

Since $\sum_{i=1}^{M}\sum_{j=1}^{n=M^i}(v_n^i(j) - \mu(v_n^i))$ is always 0, we get:

$$RSS' = \sum_{i=1}^{M}\sum_{j=1}^{n=M^i}(v_n^i(j) - \mu(v_n^i))^2 + \sum_{i=1}^{M}(\mu(v_n^i) - v_c'(i))^2$$

On the other hand, $\sum_{i=1}^{M}\sum_{j=1}^{n=M^i}(v_n^i(j) - \mu(v_n^i))^2$ depends only on numerical values, which do not vary no matter what transformations are done. So the minimization of the *RSS'* can be simplified to the minimization of $\sum_{i=1}^{M}(\mu(v_n^i) - v_c'(i))^2$

ACKNOWLEDGMENTS

Partial support was provided by NSF grants 1050477, 0959979, 0937854, and 1117132 and by a Brookhaven National Lab LDRD grant. Additional support was rendered by the IT Consilience Creative Project through the Ministry of Knowledge Economy, Korea. Finally, we are also indebted to an anonymous reviewer in a previous submission for giving us some valuable advice on statistics.

REFERENCES

[1] Alpern, L. Carter, "The Hyperbox" *Proc. IEEE Visualization*, pp. 133–139, 1991.

[2] J. Bertin. *Semiology of Graphics: Diagrams, Networks, Maps*. University of Wisconsin Press, 1983.

[3] A. Biswas, S. Dutta, H. Shen, J. Woodring, "An information-aware framework for exploring multivariate data sets," *IEEE Trans. on Visualization and Computer Graphics*, 19(12): 2683-2692, 2013.

[4] C. Chen, C. Wang, K. L. Ma, A. Wittenberg. "Static correlation visualization for large time-varying volume data," *Proc. IEEE Pacific Visualization*, pp. 27-34. 2011.

[5] J. Claessen, J. van Wijk, "Flexible linked axes for multivariate data visualization," *IEEE Trans. on Visualization and Computer Graphics*, 17(12): 2310-2316, 2011.

[6] J. Cohen, P. Cohen, S. West, L. Aiken. *Applied Multiple Regression Correlation Analysis for the Behavioral Science (3rd ed.)*. Routledge, 2002.

[7] B. Ferdosi. J. Roerdink, "Visualizing high-dimensional structures by dimension ordering and filtering using subspace analysis," C*omputer Graphics Forum*, 30(3):1121–1130, 2011.

[8] M. Ghoniem, J.-D. Fekete, P. Castagliola. "On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis" *Information Visualization*, 4(2):114-135, 2005.

[9] J. Hartigan. "Direct clustering of a data matrix," *Journal of the American Statistical Association*, 67 (337):123-129. 1972.

[10] N. Henry, J.-D. Fekete. "MatrixExplorer: a dual-representation system to explore social networks," *IEEE Trans. on Visualization and Computer Graphics*, 12 (5), 677–684 2006.

[11] P. Hoffman, G. Grinstein, K. Marx, I. Grosse, E. Stanley, "DNA visual and analytic data mining.," *Proc. IEEE Visualization*., pp. 437-441. 1997.

[12] M. Huang, P. Eades, J. Wang. "On-line animated visualization of huge graphs using a modified spring algorithm". *Journal of Visual Languages and Computing*, 9(6):623-645, 1998.

[13] A. Inselberg, B. Dimsdale, "Parallel coordinates: A tool for visualizing multi-dimensional geometry," *IEEE Visualization*, pp. 361-378, 1990.

[14] S. Johansson, M. Jern, J. Johansson. "Interactive quantification of categorical variables in mixed data sets," *Proc. Conf. on Information Visualization*, pp. 3-10. 2008.

[15] J. Li, J.-B. Martens, J. van Wijk, "Judging correlation from scatterplots and parallel coordinate plots," *Information Visualization*, 9(1):13–30, 2010.

[16] S. Ma, J. Hellerstein. "Ordering categorical data to improve visualization," *Proc. IEEE Information Visualization*, pp. 15-17, 1999.

[17] M. Meyer, H. Lee, A. Barr, M. Desbrun, "Generalized Barycentric Coordinates on Irregular Polygons," *Graphics Tools*, pp. 1086-7651, 2002.

[18] J. Nam, K. Mueller, "TripAdvisorN-D: A Tourism-Inspired High-Dimensional Space Exploration Framework with Overview and Detail," *IEEE Trans. Visualization and Computer Graphics*, 19(2): 291-305, 2013.

[19] W. Peng, M. Ward, E. Rundensteiner, "Clutter reduction in multi-dimensional data visualization using dimension reordering," *Proc. IEEE InfoVis*, pp. 89–96, 2004.

[20] H. Qu, W. Chan, A. Xu, K. Chung, K. Lau, P. Guo, "Visual analysis of the air pollution problem in Hong Kong," *IEEE Trans. on Visualization and Computer Graphics*, 13(6):1408-1415, 2007.

[21] K. Olsen, R. Korfhage, K. Sochats, M. Spring, J. Williams, "Visualization of a document collection with implicit and explicit links," *Information Processing and Management*, 29(1):69-81, 1993.

[22] G. Rosario, E. Rundensteiner, D. Brown, M. Ward, "Mapping nominal values to numbers for effective visualization," *Information Visualization*, 3(2): 80-95, 2004.

[23] P. Royston, D. Altman, W. Sauerbrei. "Dichotomizing continuous predictors in multiple regression: a bad idea," *Stat Med*, 25:127-141, 2006.

[24] J. Seo, B. Shneiderman. "A rank-by-feature framework for interactive exploration of multidimensional data," *Information Visualization*, 4(2): 96-113, 2005.

[25] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," *IEEE Symposium on Visual Languages*, pp. 336–343, 1996.

[26] H. Siirtola, E. Mäkinen. "Constructing and reconstructing the reorderable matrix". *Information Visualization*, 4(1):32-48. 2005.

[27] S. Stevens. "On the theory of scales of measurement". *Science, New Series*, 103(2684), pp. 677-680, 1946.

[28] J. Sukharev, C. Wang, K. L. Ma, A. Wittenberg. "Correlation study of time-varying multivariate climate data sets," Proc. *IEEE Pacific Vis*, pp. 161-168, 2009.

[29] C. Turkay, P. Filzmoser, H. Hauser. "Brushing dimensions - a dual visual analysis model for high-dimensional data," *IEEE Trans. on Visualization and Computer Graphics*, 17(12): 2591–2599, 2011.

[30] H. Wainer. "Finding what is not there through the unfortunate binning of results: The Mendel effect," *Chance*, 19(1):49-56, 2006.

[31] S. West, L. Aiken, J. Krull. "Experimental personality designs: Analyzing categorical by continuous variable interactions," *Journal of Personality*, 64, 1–49. 1996.

[32] J. Yang, D. Hubball, M. Ward, E. Rundensteiner, W. Ribarsky. "Value and Relation Display: Interactive visual exploration of large data sets with hundreds of dimensions," *IEEE Trans. on Visualization and Computer Graphics*, 13(3):494 –507, 2007.

[33] Z. Zhang, K. T. McDonnell, K. Mueller. "A network-based interface for the exploration of high-dimensional data spaces." *Proc. IEEE Pacific Vis*, pp. 17-24, 2012.

[34] Auto MPG dataset, http://archive.ics.uci.edu/ml/datasets/Auto+MPG

[35] Automobile dataset, http://archive.ics.uci.edu/ml/datasets/Automobile

[36] College Prowler (accessed 9/09), http://collegeprowler.com

[37] US News Best Colleges (accessed 9/09), http://colleges.usnews.rankingsandreviews.com

Zhiyuan Zhang earned his PhD degree in computer science at Stony Brook University in 2014. He is currently a research scientist at Facebook. His research interests are visual analytics and information visualization, with a special focus on high dimensional data visualization, correlation analysis, and healthcare informatics. He was awarded the IBM PhD Fellowship for 2013-2014. For more information, see http://www.cs.sunysb.edu/~zyzhang/

Erez Zadok completed his PhD in Computer Science from Columbia University in 2001. He directs the File Systems and Storage Lab (FSL) at the Computer Science Department at Stony Brook University. His current research interests are file systems and storage, operating systems, energy efficiency, performance and benchmarking, security, and networking. Zadok is the recipient of the SUNY Chancellor's Award for Excellence in Teaching, the NSF CAREER Award, two NetApp Faculty awards, and two IBM Faculty awards. For more information, see http://www.cs.stonybrook.edu/~ezk/

Kevin T. McDonnell received his BS, MS, and PhD degrees in computer science from Stony Brook University in 1998, 2001, and 2003, respectively. Since 2004 he has been on the full-time faculty of Dowling College, where he is a tenured associate professor of computer science and mathematics. His research interests include scientific and information visualization, visual analytics and human computer interaction. He is a member of ACM and Phi Beta Kappa. For more information, see http://www.ktmcd.com/

Klaus Mueller received the PhD degree in computer science from The Ohio State University and is a professor of computer science at Stony Brook University. His current research interests include visualization, visual analytics, and medical imaging. He won the NSF CAREER award in 2001 and the SUNY Chancellor's Award for Excellence in Scholarship and Creative Activity in 2011. He has authored more than 160 peer-reviewed papers. He is currently the chair of the IEEE Technical Committee on Visualization and Computer Graphics and a senior member of the IEEE. For more information, see http://www.cs.sunysb.edu/~mueller