A Network-Based Interface for the Exploration of High-Dimensional Data Spaces

Zhiyuan Zhang [1], Kevin T. McDonnell [2], and Klaus Mueller [1]

[1] Visual Analytics and Imaging Lab, Computer Science Department, Stony Brook University, NY, USA

[2] Department of Mathematics and Computer Science, Dowling College, NY, USA

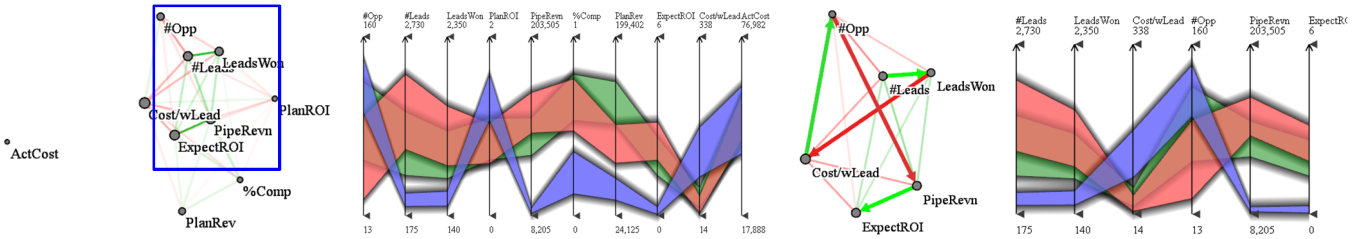


Fig. 1: Storyboarding and storytelling with the Sales dataset. (a) Original dimension network display laid out by data correlations, along with automatically computed optimal route. (b) Linked parallel coordinate display [17] with axis order determined by the route in (a). (c) The user zooms into the network (blue rectangle in (a)) and manually specifies a route that seems to best capture the *story* – the strategic model of winning the most customers (see Section 5 for more detail), (d) Linked parallel coordinate display with updated axes ordering according to the route of (c).

ABSTRACT

The navigation of high-dimensional data spaces remains challenging, making multivariate data exploration difficult. To be effective and appealing for mainstream application, navigation should use paradigms and metaphors that users are already familiar with. One such intuitive navigation paradigm is interactive route planning on a connected network. We have employed such an interface and have paired it with a prominent high-dimensional visualization paradigm showing the N-D data in undistorted raw form: parallel coordinates. In our network interface, the dimensions form nodes that are connected by a network of edges representing the strength of association between dimensions. A user then interactively specifies nodes/edges to visit, and the system computes an optimal route, which can be further edited and manipulated. In our interface, this route is captured by a parallel coordinate data display in which the dimension ordering is configured by the specified route. Our framework serves both as a data exploration environment and as an interactive presentation platform to demonstrate, explain, and justify any identified relationships to others. We demonstrate our interface within a business scenario and other applications.

Keywords: Visual analytics, parallel coordinates, multivariate data, correlation, network-based, linked displays

Index Terms: H.5.2 [Information Interfaces and Presentation]: User Interfaces - Graphical user interfaces (GUI), I.3.6 [Computer Graphics]: Methodology and Techniques - Interaction Techniques

1 INTRODUCTION

In the era of information explosion, the ultimate goal for data analysts and researchers is to understand the data collected and make useful decisions from them. More often than not, the most valuable insight comes from intricate inter-relationships among data attributes, and extracting these relationships requires skills in high-dimensional data understanding. Unfortunately, high-dimensional space exceeds human comprehension, and so, effective tools are needed to boost these capabilities. It is well known that graphical displays and visualization can show data patterns more clearly than can plain text and numbers. However, because there is no simple mapping of the multiple dimensions to a two-dimensional screen, more sophisticated visualization techniques than the arsenal of standard plots are needed.

One display that is often used for high-dimensional data visualization is the method of parallel coordinates [12]. It shows the raw data attributes on parallel axes in a sequential fashion. Parallel coordinates are good for presenting overviews of the whole, raw data set, as well as for showing relationships among the dimensions. However, the ordering of the dimensions in the parallel coordinate display has a great impact on the visual relationship analysis because the serialization of the data attributes makes it difficult to discern inter-dependencies among non-neighboring axes. As a result, a good dimension ordering can express inter-relationships clearly to the users, while poor dimension orderings can hide relationships potentially of interest. This has already been shown in early work by Bertin [4].

In the worst case, users are faced with the task of trying every possible axis ordering and remembering their findings along the way. Given n dimensions, there are a total of $n!$ possible dimension orderings. But the situation is probably not as dire as that – if we assume that a user can get a good sense of local relationships across 4 parallel axes, then we get $n!/((n-4)!4!)$ which for n=10 amounts to 210 orderings – still a daunting number.

Hence, what is sorely needed is an effective interface that can guide users in the selection of promising dimension orderings. Here it is important to note that a number of researchers have proposed possibly tunable metrics computed on the data that can suggest good dimension orderings *a priori* [1][8][14][28][29]. While this alleviates some of these problems, it does not com-

[1] E-mail: {zyzhang, mueller}@cs.stonybrook.edu
[2] E-mail: mcdonnek@dowling.edu

IEEE Pacific Visualization Symposium 2012
28 February - 2 March, Songdo, Korea
978-1-4673-0866-3/12/$31.00 ©2012 IEEE

pletely eliminate them because there is rarely just one perfect dimension ordering. Especially in interactive data exploration tasks, which are the core of visual analytics, users should be able to freely navigate the space of promising dimension orderings, and for this they need a suitable "map" of the data landscape.

To address these needs, we describe a network-based interface coupled with an interactive parallel coordinates view for exploring inter-dimensional relationships. It implements the dimension ordering interface as an interactive route planner, operating in a data-informed network where vertices represent dimensions and edges connecting two vertices represent promising dimension pairings. Users can change the route via mouse clicks, and the interface then computes a new optimized route that includes the changed route portion. Other capabilities include multi-resolution navigation and zooming of the network display.

In our paper, Section 2 presents related work. Section 3 gives a system overview. Section 4 provides a detailed description of our framework illustrated by examples. Section 5 shows its application as a platform for interactive story telling with data. Section 6 has an evaluation. Section 7 offers conclusions and future work.

2 Related Work

There are three main strategies to visualize high-dimensional data. One may embed the data into a lower dimensional space (1D-3D) using methods such as multi-dimensional scaling (MDS) [16][24], which is essentially a dimension reduction technique. MDS is widely used since it groups similar data items close together, given some distortion, and so provides a good overview onto the data. Its main drawback is that the dimension information is lost. Conversely, scatterplot matrices (SPLOM) [11] provide a simple, familiar, and clear view of the data distributions. However, due to their distributed 2D tiled layout, it can be difficult to discern relationships that extend across and involve more than two variables. Finally, parallel coordinates [12] show the raw high-dimensional data points as polylines spanning across a set of parallel vertical axes, each representing one dimension. Parallel coordinates unify the advantages of SPLOMs and MDS: they can convey all data dimensions in a single display, and they also preserve the original data dimensions. However, as mentioned, the information that can be visually extracted is highly dependent on the ordering of the dimensions. Further, clutter of the polyline display also presents problems for readability. A number of techniques have been proposed to alleviate the clutter: using free-form curves in place of the polylines [10][17][25][30], performing clutter reduction via density analysis [18], or using illustrative rendering techniques to simplify the display with a more aesthetic appearance [17].

The various paradigms can also be integrated. Schmid and Interberger [21] combine scatterplot matrices, parallel coordinates, permutation matrices, and curve display together. Wong et al. [27] combine and link parallel coordinates with scatterplots matrices, and Yuan et al. [30] integrate parallel coordinates with scatterplots, using MDS to convert multiple axes into a single subplot.

Dimension ordering in parallel coordinates has been studied for quite some time. Many methods do this fully automatically. Inselberg and Avidan [13] devise a classifier for both dimension selection and ordering. Ankerst et al. [1] define a similarity metric (either partial or global) that compares two dimensions by the RMS distance of all data points, and then optimize an ordering by ways of an approximate traveling salesman (TSP) solver – here an ant colony optimization. Conversely, Tatu et al. [23] propose a similarity-based function based on Hough Space transforms. Johansson and Johansson [14] define a weighted metric that rates dimensions by their importance with regards to correlation, outlier, and subspace cluster importance and use this rating to select relevant parallel coordinate dimensions. Similar to Artero et al. [3] they then find correlation of dimension pairings to optimize this ordering. Dasgupta and Kosara [6] devise several metrics based on the appearance of the polyline display to find good dimension orderings. Peng at al. [18] determine an ordering that minimizes clutter and outliers between adjacent dimensions. Finally, Ferdosi and Roerdink [8] order the dimensions according to high-dimensional structures identified by sub-space clustering.

A framework that in sprit is more closely related to ours is that of Fua et al. and Yang et al. [9][28][29]. They first use hierarchical dimension filtering in conjunction with dimension clustering via Ankerst's metric, but then allow users to interactively navigate the hierarchy to produce a preferred ordering. Likewise, we also aim for a framework that couples automated analysis with interaction, enabling users to make informed decisions in the ordering of the parallel coordinate dimensions. As such, our method also uses correlation as a metric to gauge dimension similarity, but it differs from existing work by our intuitive and interactive network-based framework that allows for easy navigation of the correlation matrix defined by the data dimensions. While Qu et al. [19] also visualize correlated dimensions in terms of a network layout, their system does not support a routing interface for dimension navigation, nor does it plot additional informative data characteristics on the network, as does our system. Elmquist et al. [7] apply Ankerst's metric to arrange scatterplot matrix tiles into a 2D gridded layout and then visualize the transition across tiles with 3D plots. As such, the grid functions as a map, where the routes travel along the horizontal and vertical grid lines.

Finally, Brodbeck and Girardin introduce the parallel coordinate tree [5] for presenting hierarchical and multidimensional data using a tree representation. Our framework also supports interactive multi-scale dimension exploration, but the hierarchy is seamlessly integrated into the network display.

3 System Overview

Our visual exploration framework uses two coordinated displays: a parallel coordinates display and a network display. Operations in either display are reflected in the other. The network display provides an overview of all the dimensions in terms of the pairwise correlations, whereas the parallel coordinates display shows the raw data with sequentially ordered dimensions.

In the network display, vertices correspond to dimensions and are rendered as filled circles. A vertex's size is initially determined by the corresponding dimension's coefficient of variance, but users may interactively resize vertices as desired. The vertices are laid out via a mass-spring model in which spring rest length is a function of the pairwise Pearson's correlation coefficient. As such, the layout can help users to gain a quick understanding of the correlations existing in the data by comparing the distances of the nodes representing the variables. On the other hand, the placement of the vertices and the edges drawn between them then provide the guidance for navigation and exploration.

An optimal route through the network is computed via an approximate TSP solver – we use a genetic algorithm – and this route determines the order of the dimensions given in the parallel coordinates display. It seeks to place strongly correlated dimensions next to each other. This route is set as the default, but users can re-route using simple interactive mechanisms for specifying which dimensions should be put adjacent. The parallel coordinate display changes the dimension ordering in response to user interaction in the network. Finally, multi-resolution viewing will only show networks involving large vertices.

The default route computed by the TSP solver is particular useful for novice users not familiar with the overall data scenario. They often do not have a clear understanding of all the dimensions and the variables they represent, and thus, given a data set, do not know where and how to start the exploration. The automatic ordering seeks to maximize the correlation that the sequential

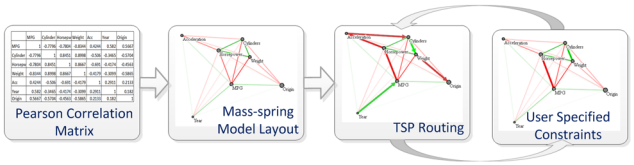


Fig. 2: The network generation pipeline.

ordering in the parallel coordinate display could provide. Then based on this ordering, users can interactively assign some constraints in the network display or directly drag and drop axes to change the ordering in the parallel coordinates display manually.

Finally, we note that the contribution of our work is not the use of TSP for automated dimension ordering in parallel coordinates – this is a fairly standard solution (see e.g. the early work of [1]). Rather, the novel aspect of our work is the network-based route planning framework that allows an informed, interactive, and user-driven dimension ordering in a visual analytics context.

4 Approach

In the following we describe the various components of our framework and the user interactions they allow. We illustrate each facet with examples derived with the following three datasets:

- Cars dataset [31] – 392 cars and 7 attributes: MPG, #cylinders, horsepower, weight, acceleration time, year and origin.
- Global Seawater Oxygen-18 dataset [22] **Error! Reference source not found.** – 25,476 data points and 8 attributes: longitude, latitude, month, year, depth, temp, salinity, and d18O.
- Synthetic dataset – 1,000 data points and 25 attributes with several attributes that behave very similarly.

We make use of Pearson's correlation coefficient in a number of ways in our work. Correlation is a statistical technique that can show whether and how strongly pairs of variables (dimensions) are related. Pearson's correlation coefficient between two variables x and y is defined as follows:

$$r_{xy} = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{X_i-\mu_x}{\sigma_x}\right)\left(\frac{Y_i-\mu_y}{\sigma_y}\right)$$

Here, μ_x and μ_y are the means, and σ_x and σ_y are the standard deviations. r ranges from -1.0 to +1.0. The closer r is to +1 or -1, the more closely the two variables are linearly related, while r close to 0 means there is no linear relationship between the two variables.

The network is then built as follows (see Fig. 2):

1. Compute Pearson's correlation for each pair of dimensions.
2. Layout the points using a mass-spring model. The forces between dimensions are computed from the correlations – highly correlated dimensions will be placed close to one another.
3. Determine a default route that covers all vertices by solving TSP approximately.

4.1 Network Construction

The vertices v in the network are positioned using a force-directed layout scheme. Vertices correspond to mass points, and graph edges are springs. The rest length of a spring is determined by the correlation coefficient of the two dimensions represented by the spring (i.e., a strong correlation corresponds to a short rest length).

In the interest of simplicity and efficiency, we employ an explicit integration method to drive the simulation. The new state of the system is determined based on the immediately preceding state of the mass-spring network. The accelerations $\ddot{v}$ of the vertices are calculated by $\ddot{v} = f/m$, where the force f is computed using Hooke's Law ($f = -kx$), with k being the spring constant and x being the displacement of the spring's end from its equilibrium position). The vertices are then moved to their new positions by $\dot{v}_{i+1} = \dot{v}_i + \ddot{v}_i \Delta t$ and $v_{i+1} = v_i + \dot{v}_i \Delta t$, where $\dot{v}$ denotes a vertex's velocity, subscripts denote time, and Δt is the time-step. The mass (m) of the points, spring stiffness values, time-step and other simulation parameters can be easily determined experimentally so that the simulation remains stable and converges quickly. The simulation is allowed to run until the maximal displacement of any vertex from one time-step to the next is less than a small constant – we chose 0.001% of the rest length of a spring representing a correlation coefficient of 0 (i.e., the maximum possible rest length of any spring).

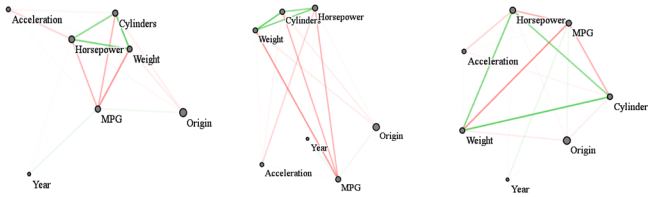


Fig. 3: Dimension layout in the *Cars* dataset via a mass-spring model. Higher color saturation on edges represents high correlation values. Green represents positive correlation, while red represents negative. (a) Layout with *absolute correlation strength*; (b) layout with *positive correlation*; (c) layout with *negative correlation preference*.

4.2 Network Semantics

The size of a network vertex encodes its significance. In statistics, if a dimension variable is more diverse, it is more interesting to drill down into. As such it potentially plays a more important role in the dataset. We use a diversity measure to encode the significance of a dimension. There are several ways to encode diversity, such as range, standard deviation, or coefficient of variation. We chose the coefficient of variation as the default because it is a normalized and comparative measure of dispersion of the distribution. However, users may choose any of these metrics, build new metrics, or alter the significance manually according to expertise.

The coefficient of variation is computed as: $c_v = \sigma / |\mu|$, where σ is the standard deviation and μ is the mean of the corresponding dimension. Significance is visually encoded as the vertex radius so that more significant dimensions are represented as larger vertices, whereas less significant ones are encoded with smaller radii.

Conversely, the edge significance is weighted by the correlation between the two dimensions linked by the edge. The correlation is encoded by color. Green encodes a correlation value of +1 while red encodes correlation value of -1, and 50% gray is used to encode 0. Linear interpolation computes the colors in between. We provide three correlation functions – correlation strength, and positive and negative correlation – to help users to comprehensively understand various dimension relationships. Fig. 3 provides illustrations using the cars dataset as an example:

(i) Correlation strength (see Fig 3a): Here the absolute value of the correlation is used to compute the rest length. Thus, high correlations correspond to shorter rest lengths. In this way, highly correlated dimensions will be drawn towards each other, whereas weakly correlated dimensions will repel each other. In the parallel coordinate display, these strongly correlated dimensions will likely be arranged next to each other by the TSP router.

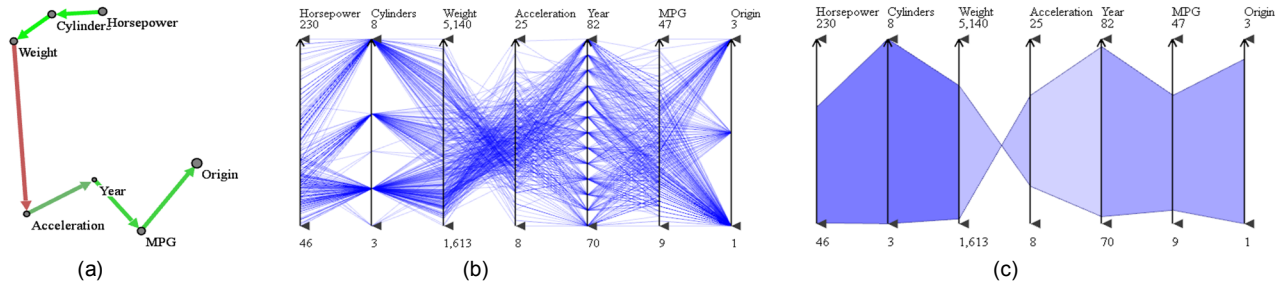


Figure 4: (a) A tour of the *Cars* dataset. In the network display, the dimension ordering is given as a series of directed edges color-coded by correlation value. (b) Parallel Coordinate display. The dimension triples (*Horsepower*, *Cylinder*, and *Weight*) and (*Year*, *MPG*, and *Origin*) are put next to each other in the parallel coordinate display because they are strongly correlated with each other. These strong positive correlations can be seen on the network display also. (c) Correlation display colors the outlines of *line bundles* (see [17]) in terms of their correlation strength (less saturation maps to lower correlation). We can also discern negative correlations by the characteristic bow-tie shapes.

(ii) Positive correlation (see Fig. 3b): Here a simple linear transformation of the correlation value, $(r_{XY}+1)/2$, is used, where r_{XY} represents the correlation between dimensions X and Y. Now, positively correlated dimensions will tend to be drawn towards each other, whereas negatively correlated dimensions will repel.

(iii) Negative correlation (see Fig 3c): This is the opposite case of the positive correlation preference. Now $(-r_{XY}+1)/2$ is used to compute the rest length.

Thus, since the vertices are laid out by functions of correlation, depending on which function the user chooses, the user can learn quickly which dimension pairs have what type of correlation. For example, if vertices are laid out by strength of correlation (absolute value of correlation), then close dimensions have strong correlations while far-away dimensions have weak correlations. The edge's color then reveals whether it represents a positive correlation (green) or a negative one (red). Similar insight emerges from the positive/negative correlation-preferred layout. This display is helpful since the parallel coordinate display makes it difficult to discern correlation information if two dimensions are not adjacent to each other – the network view readily solves this problem.

4.3 Network-Driven Dimension Ordering

As mentioned, our approach conceives a ordering of the parallel coordinates dimension axes by specification of a path in the correlation-based network display. The shortest path covering all vertices is then one that maximizes the amount of correlation exposed by the parallel coordinate display. Finding such a path is the well-known TSP problem. We have chosen a genetic-algorithm-based TSP solver since it defines a maximum time bound for computing the solutions. Users can modify this time bound to strike a balance between performance and accuracy. In our implementation, we set the time bound to 1s which makes the system sufficiently responsive for interactive exploration. We found that when the number of dimensions n<30 the routing is reasonably accurate. Conversely, although the performance is dependent both on the number of vertices and the vertex locations, for n>30 with the given time bound of 1s, the solution is sometimes not accurate. Here, according to our experiments, an edge-length-based greedy algorithm typically produces better results at a much reduced time. We thus define a threshold d=30. When n<=30 we use a genetic algorithm while for n>30 we run the greedy algorithm.

Fig. 4a shows a route within the network display of the Cars dataset and Fig. 4b shows the associated parallel coordinate display with the dimension ordering implied by the route. The experienced reader will notice that the correlations revealed by the line structure of the parallel coordinate display are quite similar to those visualized by the vertex distances and edge colors in the network display. Of course, the latter is much easier to discern for less experienced users.

To aid these inexperienced users in the visualization of correlations also within the parallel coordinate display, we have devised mechanisms that add additional illustrative hints. These techniques are inspired by earlier work of a subset of this article's authors [17]. First, a bounding hull of the line bundles is computed based on the line centers and standard deviations. The difference now is the bounding hull we use for negative correlated dimensions. If two dimensions are negatively correlated, instead of using a band-shape, the characteristic bow-tie shape is employed. Then the bounding hull is colored in terms of their correlation strength where less saturation maps to lower correlation. Fig. 4c shows an example for this scheme. We observe that the coloring maps nicely to the vertex distances in the network display.

4.4 Interactions with the Network

To be effective and appealing for mainstream application, interactions should use paradigms and metaphors that users are already familiar with. For this reason, we employ a route planning paradigm that resembles those found in Web-based, interactive maps.

Our interface allows users to interactively assign constraints to modify the route for data exploration by simple mouse interactions. Such constraints include specifying edges that should to be maintained or avoided on the route, as well as vertices that should be avoided. If an edge is marked to be included in the path, then all paths that do not pass through the edge will be penalized. Conversely, if an edge is marked to be avoided, then all paths that pass through this edge will be penalized. If a vertex is marked to be avoided, then we remove the vertex from the TSP computation. Every time a user makes modifications of this nature the TSP algorithm is rerun to produce a new optimal ordering observing these constraints. Finally, users are also able to specify at which dimension the route starts. In this case, the TSP-generated paths will contain only those starting with the specified dimension.

It can sometimes be useful to duplicate a dimension in the parallel coordinate display to visualize its interaction with two different variable pairs at the same time. We support such orderings by allowing users to specify a *loop constraint* on the TSP path. For example, a user might be interested to examine variable 2 with variables 1 and 3 but also with variables 4 and 5. One of these pairs will then form a loop and the other will be part of the regular path. To identify the best configuration we run TSP multiple times. In our example, one possible outcome might be that variables 1, 2, and 3 are part of the path and variables 4 and 5 form the loop centered at variable 3, but the opposite can also be the result. This mechanism extends to multiple loop constraints.

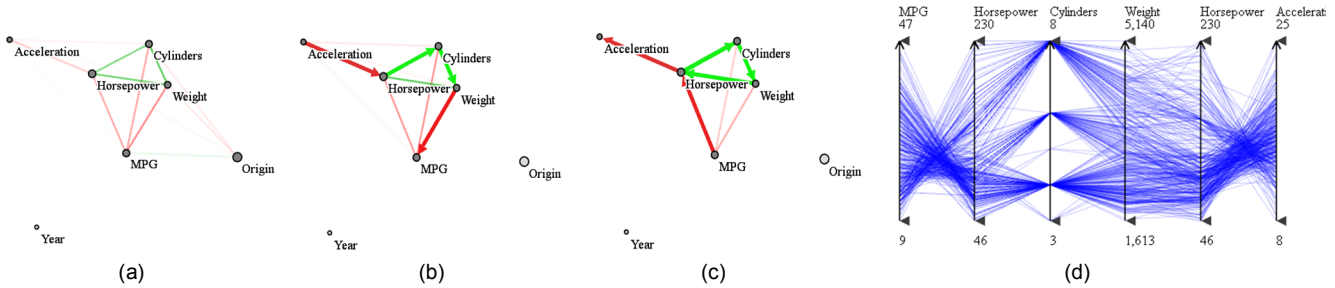


Fig. 5: User interaction and constraint imposition for the *Cars* dataset. (a) Network Display with *Year* being filtered out by multi-scale zooming. (b) The route is edited to avoid unrelated dimensions *Year* and *Origin*. (c) Routing with a cyclical constraint. Dimension *Horsepower* is visited more than once, which makes it adjacent to 4 dimensions (*MPG*, *Weight*, *Cylinders*, and *Acceleration*). The corresponding parallel coordinate display is shown in (d), where we can see that *Horsepower* has been duplicated.

Finally, users can also modify the significance of the vertices via a *significance slider*. This slider controls the size of the vertices that are taken into account for the routing. Extending the scale ignores smaller vertices both in the network and for the auto-routing. The ignored dimensions will then be removed from the parallel coordinate display accordingly. A similar mechanism controls the inclusion of insignificant edges – edges with lower correlations strengths – into the TSP calculations.

Fig. 5 shows some snapshots of such an interactive routing session, again using the Cars dataset. First, the user enacts the significance slider to filter out the less significant attributes (the attribute *Year* in this case), as shown in Fig. 5a. The user then observes that *Origin* appears to be uncorrelated with (i.e., is distant from) the remaining attributes. So he removes it from the route (Fig. 5b) to yield a more compact parallel coordinate display (not shown). Further, he also observes that *Horsepower* is highly correlated with both *#Cylinders*, and *Weight* (but also negatively correlated with *MPG* and *Acceleration*). In order to see all of these relationships conveniently in the parallel coordinate display he adds a loop to the route, interspersing *Horsepower* between both relationships and also visualizing the strong 3-way relationship in the loop (Figs. 5c and 5d).

4.5 Multi-Scale Zooming in the Network Display

As described thus far, our approach readily supports around 20 dimensions without causing too much clutter in the network display. This is commonly referred to as *multidimensional visualization*. In order to support a visualization of high-dimensional datasets, both in the network display as well as in the parallel coordinate display, we require a multi-scale framework. We provide this functionality via a zooming interface.

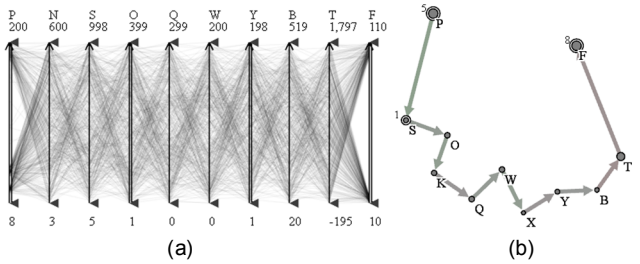


Fig. 6: Zooming out in the Network Display of a 25-dimensional synthetic dataset. Representative dimensions (*F, P, S*) are displayed by double lines in the parallel coordinates display (a) and double circles in the network display (b). Also, the number of dimensions hidden by each representative dimension is given by the small number in the upper left corner of its vertex in the network display.

We build a hierarchy of the dimensions for multi-scale zooming. This requires a similarity metric. Yang et al. [29] describe an approach that uses similarity-based metrics to decide whether similar dimensions should be merged or not. In our work, we use a correlation-related metric for this purpose. After mass-spring layout, the distances between pair of dimensions provide a global indication of this correlation. Using these distances, we then provide a zooming experience similar to that offered in popular web-based map exploration programs. As a user zooms out of the display, nearby dimensions will merge into one, and as he/she zooms back in, the merged dimensions split into the original dimensions.

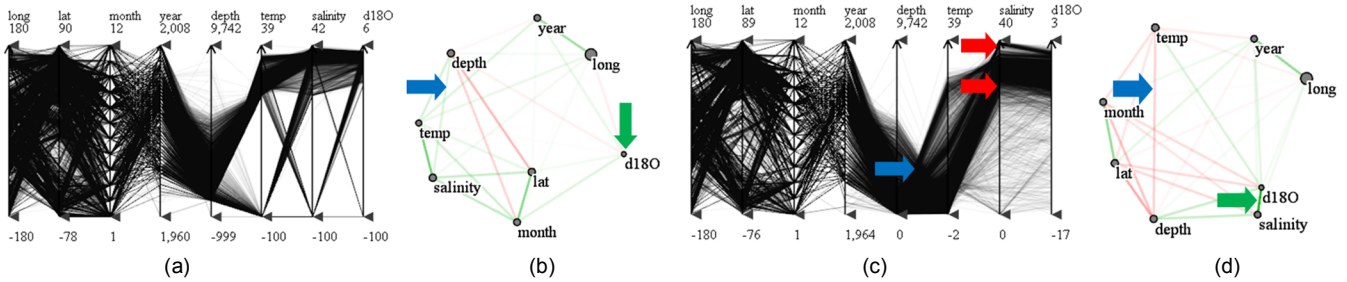
When merging close dimensions, a representative dimension is needed to represent all of the merged dimensions. There are several ways to choose a representative dimension, which is also discussed in [29]. In our system, we choose the most significant dimension as the representative one, using the mechanisms described in Section 4.2. Users can always manually control whether to merge or collapse a representative vertex by mouse-clicks. Finally, the zooming extends to the parallel coordinate display as well – which will also only show the representative dimensions.

When users zoom into the network display, some of the vertices may vanish from the screen, which practically means that users are now focusing on the dimensions remaining in sight. The parallel coordinate display will then display only the dimensions remaining on the network display to show their neighborhoods at greater detail. The dimension order is determined by the TSP which considers only the vertices shown on the current display.

To illustrate this function consider Fig. 6 which uses the synthetic dataset. This dataset has 25 dimensions in total, but showing them all will lead to a cluttered network display and a fairly extended parallel coordinate display (both are not shown). Since several dimensions behave very similarly, we can simply use one of these as a representative. Hence, in our network display, as the user zooms out, similar dimensions merge into one which significantly reduces the clutter (Fig. 6b). Likewise, the coupled parallel coordinate display shows only the remaining dimensions not abstracted away in the network display (Fig. 6a).

4.6 Effect of Parallel Coordinate Display Interactions

Our parallel coordinate display follows the now fairly standard practice of allowing users to filter out undefined values in some dimension (e.g., the undefined values are set to -999) or only visualize a subset of the data falling within a certain range interval of a dimension. For this purpose, our interface provides brush handles for each dimension (see the little triangle on top and bottom of each dimension axis in Fig. 5d) which can be used to bracket some portions of the corresponding dimension. This "bracketing" prompts the system to filter out the data points that lie outside the handles and only display the remaining data points with proper



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

Fig. 7: The effect of dimension bracketing, using the global seawater oxygen-18 dataset. (a), (b) Parallel coordinate and network display of the original dataset with undefined values in dimension *depth*, *temp*, *salinity* and *d18O*. (c), (d) After filtering out the undefined values and bracketing the dimensions. We can now see a strong positive correlation between *salinity* and *d18O* (green arrow in (d)), while in the original, unfiltered dataset with undefined values, *d18O* is incorrectly shown not to be strongly correlated with any other dimensions (green arrow in (b)).

normalization. All data attributes (e.g., correlation, variance) are then re-computed based on the remaining data.

Let us now demonstrate the effect of parallel-coordinate bracketing and filtering on the co-linked network display. As shown in Fig. 7a for the Global Seawater Oxygen-18 dataset, the original dataset contains undefined values (-100) for four dimensions: *depth*, *temperature*, *salinity*, and *d18O*. This significantly influences the correlation computation – we observe very weak correlations among all dimensions in Fig. 7b. Following, Fig. 7c shows the parallel coordinate display after bracketing the four dimensions, filtering out the undefined values and renormalizing their bottom axis brackets – they now show the true minimum values. We may also call this a zooming operation. We can now readily see in the network display (Fig, 7d) that there is in fact a strong correlation between dimension *salinity* and *d18O*.

5 Using the Network for Story Telling with Data

The parallel coordinate display provides a sequential view of the data. Reading the plot from left to right is similar to reading a story from beginning to the end. Of course, just like in parallel coordinates, readers of a book or viewers of a movie may skip back and forth to recap what has already been seen or peek ahead what is yet to come. Directors of movies commonly use story boards to organize the shots into a suitable sequence. Here one typically aims to arrange a sequence that builds the story in a coherent manner, with some sort of climax in the end. With 'coherence' one might define that subsequent scenes bear some degree of correlation (yet we admit that movies exist that have turned the lack of coherence into an art form).

We propose to utilize our network display as a means for story boarding in information visualization with parallel coordinates. Our network display provides all needed information and functionality to help visual analysts script insightful and informative 'movies' in parallel coordinates. It reveals correlated shots (i.e. dimensions) and it allows automated and manual interactive arrangement of these shots, using the network interaction facilities.

We shall demonstrate the use of this interface via our sales campaign dataset. It consists of 900 data points (one per sales person) and 10 attributes: %Completed, #Leads, Leads Won, #Opportunity, Pipeline Revenue, Expected ROI (Return on Investment), Actual Cost, Cost/WonLead, Planned Revenue, and Planned ROI. There are three pre-clustered sales teams.

Before delving into this case we need to review some basics. The typical corporate sales pipeline begins with a *lead generator* yielding a number of prospective customers with some level of probability to actually close the deal. The *leads* are fed a low-cost stimulus to probe their interest. Upon a positive response these leads are called *won leads* and receive a stepped-up sales pitch, at a *cost/won lead*. If this pitch wins further positive response then these won leads turn into *opportunities*. In practice there are many more levels, but this may serve as a sufficient model here.

As a practical scenario let us imagine a meeting of sales executives who would like to review the strategies of their various sales teams. The data contains three sales teams of a large corporation with a couple of hundred sales people in each team. Jim, one of the sales strategy analysts begins and constructs Fig. 1a (1st page). This display reveals that the #leads, #won leads, #opportunities, and cost/won lead are somewhat related. The TSP computes an initial route that gives rise to the parallel coordinate display in Fig. 1b. Jim quickly notes that this route does not really represent the actual flow of a lead through the sales pipeline and changes the route to #leads → #won leads → #opportunities (not shown). Soon after, Kate, another sales analyst in the meeting room, realizes from looking at the network display that cost/won lead is nearby and has a strong positive correlation with #opportunities but also a negative correlation with #won leads. She suspects that some insight could possibly be gained from routing these two latter variables through the first. So she uses the mouse and designs the route shown in Fig. 1c which gives rise to the parallel coordinate display of Fig. 1d. From the parallel coordinate plot it is now immediately obvious that the blue team employs a very different strategy than the green and the red teams. The blue team generates far fewer leads but spends much more resources on each which apparently gives it an advantage in the final outcome. It can also be observed that the blue team is much more consistent than the other teams, as indicated by the much narrower band.

6 Evaluation

We tested two aspects: the accuracy of the dimension ordering and the utility and effectiveness of the linked interface.

6.1 Dimension Ordering

We have compared our correlation-based TSP-ordering with other automatic ordering methods proposed in the literature, and the results are presented in Fig. 8. Here we used the synthetic dataset. The dataset has 25 dimensions (*A,B,C,…,Y*) and 1,000 data points, and is made purposely to have two subspaces – a 9D subspace (*A,C,E,F,G,H,R,U,V*) with two clear clusters and a 6D subspace (*D,I,J,L,M,P*) with three clear clusters (as shown in Fig. 8d and 8e). All others are noise dimensions. Fig. 8a shows the original ordering, and Fig. 8b uses the clutter-based ordering [18]. Though some of the structure can be observed (most noise dimensions tend to be next to each other and the presence of subspaces is apparent), the general structure is not clearly shown. Fig. 8c is using Ankerst's method [1]. The two subspaces are well captured, but the noise dimensions are split into two parts. The structure of the dataset is best represented in Fig. 8d, which is generated by Ferdosi's SBF dimension ordering [8]. The two subspaces and the

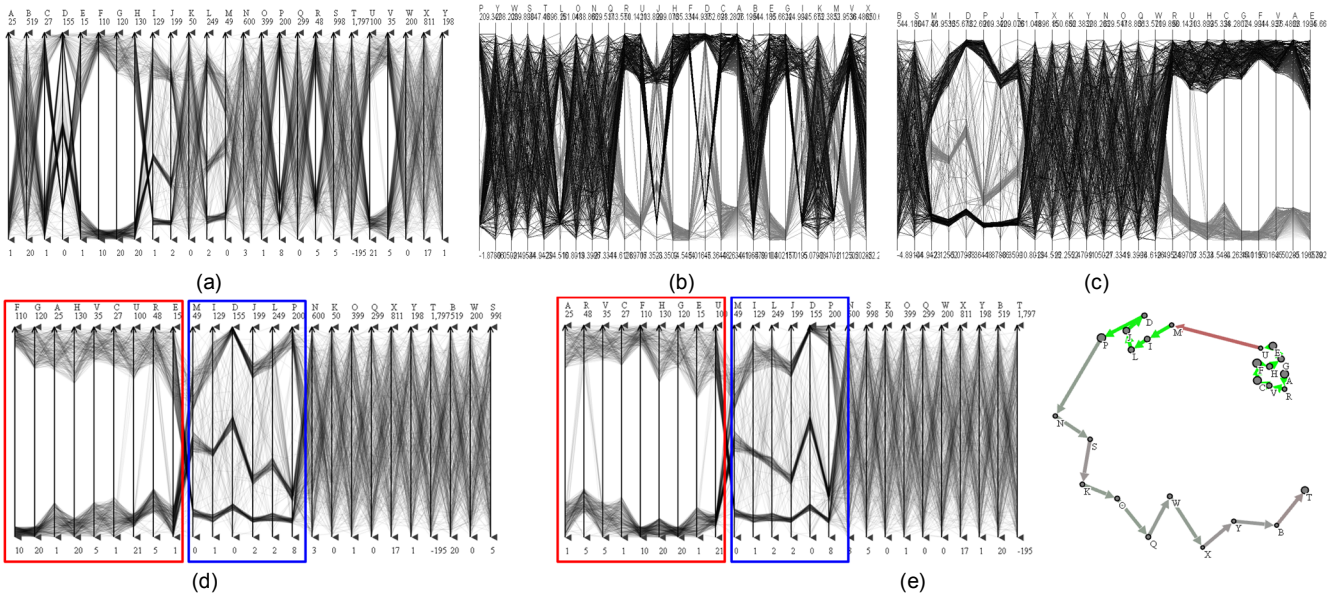


Fig. 8: Comparison of our automatic dimension ordering algorithm with other methods. (a) Original ordering. (b) The clutter-based ordering of Peng et al. [18] is unable to put the proper dimension order to show sub-clusters. (c) Ankerst's method [1] can capture the two subspaces, but the other un-related dimensions are split into two parts. Figures 8b and 8c are generated by xmdvTool [26]. (d) Ferdosi's subspace-based dimension ordering [8], which is able to capture the structure of the dataset (2 cluster subspace highlighted in red rectangle and 3 cluster subspace highlighted in blue rectangle). (e) Our method: the result is quite similar to (d).

noise dimension are well separated. Fig. 8e is obtained by our TSP-based approach. We observe in that the result is quite competitive with the SBF method, although the dimension orderings within the subspaces are different for each of the two methods. It appears that the TSP-based approach leads to dimension orderings in which the cluster values of adjacent dimensions change in a more linear fashion – compare the paths of the center cluster in the second subspace (blue window) in Fig. 8d and e.

But despite these positive observations, we wish to note that the work presented here is mainly about our interactive network-based interface and not about the quality of the TSP-based dimension ordering. More comprehensive studies would be needed to validate the latter. But as the sales campaign example in Section 5 has shown, an automated ordering might not always be useful and might require substantial edits to yield a possibly less optimal but more meaningful ordering, which our interface readily facilitates.

6.2 Interface

We aimed for a framework that can even be accessible to users with no significant prior training in high-dimensional data analysis. So we performed a user study with members of this potential user group to get more insight into the effectiveness of our framework. We used the sales dataset because it contains some specific easy-to-grasp relationships. Our goal was to see whether and how quickly the test subjects could identify the relationships described in Section 5, i.e., why the blue team behaves differently from the others (using a fewer leads and won leads to generate more opportunities). Our hypotheses for the user study were:

H1. With the help of our network-based display, users are able to find the relationship more accurately.

H2. With the help of our network-based display, users are able to find the relationship faster.

To test these hypotheses we invited 18 graduate students (none majored in business) to participate. First we spent about 20 minutes to give them an introduction to our framework. We used the Cars dataset because this domain is the most generally familiar. We made sure that after this period all subjects knew the concepts of parallel coordinates and the network display and knew all the interactions supported by our framework. Then we randomly split the subjects into two equal-sized groups: one group only used the parallel coordinate display along with the raw data table (Group1), and the other group used both displays (Group2). We then asked each subject to select the attribute in the sales dataset that best explained the scenario elaborated on in Section 5.

In Group1, 3 students found the correct answer, i.e. *cost/wonLead*. In Group2, 7 students picked *cost/wonLead* because this attribute is the closest one with a dark red edge to *#leads* and *#leadsWon*. 1 student picked 3 attributes (*cost/wonLead, pipelineRev, and plannedROI)* which are nearby and said the scenario might be caused by the combination of them (regarded as 1/3 correct). So in this case we observed 7.33 (7+1/3) students with the right answer, more than twice than in Group1. Therefore our network display clearly helped. The corresponding p-value is 0.039, which means Hypothesis 1 is confirmed.

To test Hypothesis 2, we used an independent two-sample t-test based on equal sample sizes and equal variance. On average, participants spent more time to find the answers in Group1 (*Mean* = 20.22 seconds) than those in Group2 (*Mean* = 11.56 seconds). The corresponding t-value is 2.85 and p-value=0.018. For 18 participants (degree of freedom = 16), t must be at least 2.12 to reach $p < 0.05$, so this difference was statistically significant.

Also, among the 18 students, 11 of them claimed that it was the first time they had seen a parallel coordinate display. It was interesting to notice that these 11 students asked more questions and spent more time on learning the parallel coordinate system than on the network display. They stated that the network display was quite easy to understand since they had seen similar displays before. Some mentioned that the network display reminded them of the "Get direction" feature in Google Maps. This insight suggests that our network-based navigation interface is quite accessible, even to novice users.

7 Conclusions

We have presented an interactive network-based interface coupled with a parallel coordinates display, offering users various interaction tools to control the dimension ordering and assess correlation in the data. The framework can serve both as a data exploration environment and as an interactive presentation platform to demonstrate, explain, and justify identified relationships to others. We showed that the synergy of the network and parallel coordinate display offers a great deal of analytical power to users and it also allows them to explore their datasets more efficiently. Due to its resemblance to popular routing tools used in online maps the interaction with our system is also intuitive and natural.

For future efforts, we would like to implement animated axes transitions when the user changes the ordering with the routing interface. This will make it easier for users to understand the changing relationships. Further, our system could also be used for interactive dimension reduction via subspace clustering [15]. This would create islands in the landscape. However, a potential difficulty is that dimensions may appear in more than one subspace, which is not fully captured in this isolated island paradigm.

Correlation can be affected by outliers, non-linear relationships, multicollinearity, heteroskedasticity, and multicollinearity, none of which we currently address. To gain more robustness, it would be worthwhile to implement outlier detection and/or removal algorithms, and also devise methodologies for detecting and visualizing non-linear relationships. Finally, both parallel coordinate and correlation do not work well for categorical data in general. One possible solution is the Distance-Quantification-Classing approach devised by Rosario et al. [20].

Our user-study suggests that the network interface might already represent a good stand-alone tool to visualize associations. But the many questions raised by the test subjects also suggest that parallel coordinate displays, although quite similar to line plots, still require a high level of visualization literacy – more than most people possess today. This inspires further research in adapting familiar graphic design methodologies towards this interface.

Acknowledgements

This work was funded in part by NSF grants 1050477, 0959979, 1117132, and a Brookhaven National Labs LDRD grant.

References

[1] M. Ankerst, S. Berchtold, D. Keim, "Similarity clustering of dimensions for an enhanced visualization of multidimensional data,"*Proc. IEEE InfoVis,* pp. 52-60, 1998.

[2] K. Arakawa, S. Tamaki, N. Kono, N. Kido, K. Ikegami, R. Ogawa, M. Tomita, "Genome Projector: zoomable genome map with multiple views," *BMC Bioinformatics,* 10 (31), 2009.

[3] A. Artero, M. de Olivera, H. Levkowitz, "Enhanced high-dimensional data visualization through dimension reduction and attribute arrangement," *Proc. IEEE InfoVis*, pp. 707–712, 2006.

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

[5] D. Brodbeck, L. Girardin. "Visualization of large-scale customer satisfaction surveys using a parallel coordinate tree," *Proc. IEEE InfoVis* pp. 197-201, 2003.

[6] A. Dasgupta, R. Kosara, "Pargnostics: screen-space metrics for Parallel Coordinates," *IEEE TVCG*, 16(6):1017-1026, 2006.

[7] N. Elmqvist, P. Dragicevic, J.-D. Fekete, "Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation," *IEEE TVCG*, 14(6):1141–1148, 2008.

[8] 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.

[9] Y. Fua, M. Ward. E. Rundensteiner, "Structure-based brushes: A mechanism for navigating hierarchically organized data and information spaces," *IEEE TVCG. 6*(2):150–159, 2000.

[10] M. Graham, J. Kennedy, "Using curves to enhance parallel coordinate visualizations," *Proc. IEEE Info Vis*. pp.10-16, 2003.

[11] J. Hartigan, "Printer graphics for clustering," *Journal of Statistical Computation and Simulation,* 4(3):187-213, 1975.

[12] A. Inselberg, B. Dimsdale, "Parallel Coordinates: a tool for visualizing multi-dimensional geometry," *IEEE Vis*, pp. 361–378, 1990.

[13] A. Inselberg, T. Avidan, "The automated multidimensional detective," *Proc. IEEE InfoVis*, pp. 112-119, 1999.

[14] S. Johansson, J. Johansson, "Interactive dimensionality reduction through user-defined combinations of quality metrics," *IEEE TVCG. 15*(6):993–1000, 2009.

[15] H. Kriegel, P. Kröger, A. Zimek, "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering," *ACM Trans KDD*, 3(1), 2009.

[16] J. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika,* 29(1):1–27, 1964.

[17] K. McDonnell, K. Mueller, "Illustrative Parallel Coordinates," *Computer Graphics Forum*, 27(3):1031-1027, 2008.

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

[19] H. Qu, W. Chan, A. Xu, K. Chung, K. Lau, P. Guo, "Visual Analysis of The Air Pollution Problem in Hong Kong," *IEEE TVCG*, 13(6):1408-1415, 2007.

[20] G. E. Rosario, E. A. Rundensteiner, D. C. Brown, and M. O. Ward, "Mapping Nominal Values to Numbers for Effective Visualization," *Information Visualization*, 3(2): 80-95, 2004.

[21] C. Schmid, H. Hinterberger, "Comparative multivariate visualization across conceptually different graphic displays." *Proc. SSDBM*, pp. 42-51, 1994.

[22] G.A., Schmidt, G. R. Bigg and E. J. Rohling. "Global Seawater Oxygen-18 Database - v1.21" http://data.giss.nasa.gov/o18data/ 1999.

[23] A. Tatu, G. Albuquerque, M. Eisemann, H. Theisel, M. Magnor, D. Keim, "Combining automated analysis and visualization techniques for effective exploration of high-dimensional data," *Proc. IEEE VAST*, pp. 59–66, 2009.

[24] W. Torgerson, "Multidimensional scaling: I. theory and method," *Psychometrika*, 17:401–419, 1952.

[25] H. Theisel. "Higher order Parallel Coordinates," *Proc. VMV*, pp. 415-420, 2000.

[26] M. Ward, "XmdvTool: Integrating multiple methods for visualizing multivariate data," *Proc. IEEE Visualization.*, pp. 326-333, 1994.

[27] P. Wong, R. Bergeron, "Multivariate visualization using metric scaling," *Proc. IEEE Visualization*, pp. 111-118, 1997.

[28] J. Yang, W. Peng, M. Ward, E. Rundensteiner, "Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets," *Proc. IEEE InfoVis*, pp. 105-112, 2003.

[29] J. Yang, M. Ward, E. Rundensteiner, S. Huang, "Visual hierarchical dimension reduction for exploration of high dimensional datasets," *Proc. VisSym*, pp. 19–28, 2003.

[30] X. Yuan, P. Guo, H. Xiao, H. Zhou, H. Qu, "Scattering points in parallel coordinates," *IEEE TVCG*, 15(6):1001-1008, 2009.

[31] http://archive.ics.uci.edu/ml/ (UCI Machine Learning Repository)