

# ScagExplorer: Exploring Scatterplots by Their Scagnostics

Tuan Nhon Dang\*  
University of Illinois at Chicago

Leland Wilkinson†  
University of Illinois at Chicago  
Skytree Software Inc.

## ABSTRACT

A scatterplot displays a relation between a pair of variables. Given a set of  $v$  variables, there are  $v(v-1)/2$  pairs of variables, and thus the same number of possible pairwise scatterplots. Therefore for even small sets of variables, the number of scatterplots can be large. Scatterplot matrices (SPLOMs) can easily run out of pixels when presenting high-dimensional data. We introduce a theoretical method and a testbed for assessing whether our method can be used to guide interactive exploration of high-dimensional data. The method is based on nine characterizations of the 2D distributions of orthogonal pairwise projections on a set of points in multidimensional Euclidean space. Working directly with these characterizations, we can locate anomalies for further analysis or search for similar distributions in a “large” SPLOM with more than a hundred dimensions. Our testbed, ScagExplorer, is developed in order to evaluate the feasibility of handling huge collections of scatterplots.

**Index Terms:** I.5.2 [Pattern recognition]: Design Methodology—Pattern analysis

## 1 INTRODUCTION

Because paper and computer screens are limited to two dimensions, graphical display of multivariate data is intrinsically difficult. To visualize multivariate data, we often project a higher-dimensional point cloud of data onto the plane. Consequently, it is essential to select projections that reveal important characteristics of the data. In the simplest of cases, projections can be selected using linear maps such as multiple regression or principal components. These are popular and easy to compute. By contrast, a more data-driven approach is to examine scatterplots of variables for unusual distributional features such as convexity, compactness, or outliers. Arranging these plots in a Trellis display or scatterplot matrix (SPLOM) allows one to explore individual scatterplots interactively.

Previous researchers have applied data-driven approaches to moderate-sized collections of scatterplots [26, 27, 13]. The scale of these efforts has been constrained by display limits and computational complexity. In this paper, we develop an alternative approach in order to deal with the scalability problem for SPLOMs in terms of data sets larger than one hundred dimensions. Our goal is to be able to organize these plots into meaningful subsets in reasonable time and to present these plots to users in a rich exploratory environment.

Our contributions in this paper are:

- We have proposed a way to retrieve similar scatterplots to a plot of interest-based euclidean distance in scagnostics space.
- We have implemented the leader algorithm [14] to cluster similar scatterplots. We have developed a dynamic algorithm that leverages force-directed graph methods to cluster the leader scatterplots. This cluster layout provides a comprehensive

\*e-mail: tdang@cs.uic.edu

†e-mail:leland@skytree.net

summary of the 2D relations of variables in a dataset. Users can then select a leader to investigate for further details.

- We have proposed a method for filtering scagnostics using parallel coordinates to refine subsets of scatterplots sharing common features.

The paper is structured as follows: We describe related work in the following section. Then we introduce our ScagExplorer testbed and illustrate it on real datasets. We present test results of ScagExplorer in Evaluation. In our Conclusion, we argue that our approach makes it possible to explore huge datasets without resorting to basic statistical summaries (means, standard deviations, correlations, etc.). By going beyond classic statistical summaries, we are able to deal with unusual distributions, mixtures of distributions, outliers, and other important features found in real datasets.

## 2 RELATED WORK

### 2.1 Scagnostics

Graph-theoretic scagnostics was introduced by Wilkinson [26], based on an unpublished idea of John and Paul Tukey. The implementation of scagnostics are described in detail in [27].

Scagnostics measures depend on proximity graphs that are all subsets of the Delaunay triangulation: the minimum spanning tree (MST), the alpha complex [10], and the convex hull. Figure 1 shows an example of the three geometric graphs generated on the same set of data points.

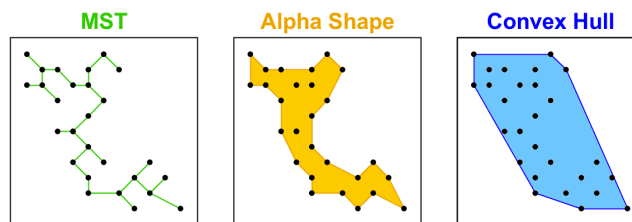


Figure 1: Graphs for computing scagnostics measures.

Figure 2 shows some example scatterplots and their scagnostics. In particular, the scatterplots with a low score on the associated scagnostic are on the left while the scatterplots with a high score on the associated scagnostic are on the right.

### 2.2 Feature-based Approaches

Seo and Shneiderman [22] computed statistical summaries (means, standard deviations, correlations, etc.) on univariate and bivariate distributions and then ranked them in order to identify similar distributions. Their Rank By Feature tool helps a viewer to navigate through a relatively large corpus of statistical data. Other researchers have developed scagnostics-type measures for parallel coordinates [9], pixel displays [21], 3D scatterplots [11], and other graphics [23, 2].

Another product of Shneiderman’s lab, Time Searcher [15], introduced a suite of features designed to detect recurring or unusual patterns in time series data. Some of these features, such as sudden increases and decreases in amplitude, proved useful for scanning relatively long financial series. Shape Search Edition of Time

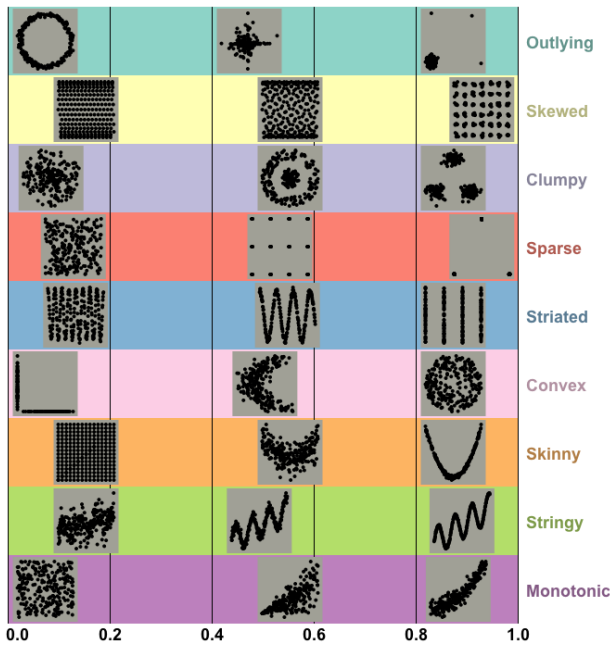


Figure 2: Example scatterplots and their scagnostics measures.

Searcher [12] provides a set of shapes and the attributes by which time series can be identified, compared, and ranked.

Yang et al. [28] proposed a *Value and Relation (VaR)* technique explicitly conveying the relationships among the dimensions of a high-dimensional dataset based on the data values in each dimension. Data values are first normalized and binned within each dimension. Then, the distance matrix is computed on binned data. The VaR technique helps users grasp the associations among dimensions (similar to the Monotonic measure) as depicted in Figure 3(a). However, this technique fails to capture more complicated relations as depicted in Figure 3(b) and Figure 3(c) that can be captured by the Stringy and Striated measure.

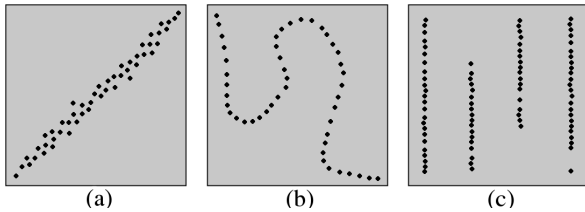


Figure 3: a) Monotonic distribution b) Stringy distribution c) Striated distribution.

TimeSeer [8] uses scagnostics for organizing multivariate time series and for guiding interactive exploration through high-dimensional data. TimeSeer consists of 2 systems: a SPLOM viewer and a time series viewer. The SPLOM viewer provides guidance for selecting interesting pairs of variables. Then, the time series viewer can graph scagnostics time series of up to 10 selected pairs of variables in a single display.

### 2.3 Subspace Analysis

To deal with the “Curse of Dimensionality”, subspace analysis techniques examine subsets of dimensions instead of the entire space. This comes from the expectation that data patterns are prominent only in a few dimensions for most high-dimensional

data. Consequently, many subspace analysis techniques sacrifice completeness for simplicity (to obtain sub-linear time complexity).

Subspace clustering [1] is an extension of feature selection that aims to detect clusters and a set of relevant dimensions for each cluster. A review on various subspace clustering algorithms can be found in [20]. More recent subspace analysis approaches have been designed for users to navigate through the subspaces [24], to find interesting low-dimensional projections [4], to interpret the result of subspace clustering [25], and to analyze subspace cluster characteristics in-depth [6].

## 3 SCAGEXPLORER OVERVIEW

ScagExplorer is a testbed we developed in order to evaluate the feasibility of handling huge collections of scatterplots. Unlike the original application outlined in [27], we designed ScagExplorer to deal with a large scatterplot space spanning thousands of scatterplots. Furthermore, ScagExplorer does not *require* the use of a scatterplot matrix. Instead, it uses a novel force-directed layout of scatterplots plus brushing, linking, and details on demand to organize a much larger number of scatterplots than can be handled in a SPLOM or Trellis display. We accomplish this by introducing a method for processing exemplar scatterplots that reduces the number of plots that must be considered.

### 3.1 Dissimilarity of Two Scatterplots

Why is scatterplot similarity a relevant and important concept and/or starting point when exploring data?

Answer 1: Figure 4 shows life expectancy of male vs. female in three years from 1982 to 1984. Each data point is a country. The three outliers in these plots are Iran in red, Iraq in blue, and El Salvador in purple. The Iran-Iraq war (First Persian Gulf War) lowered the life expectancy of males because men were needed for the war. A similar situation happened to El Salvador because this was the time period inside the Salvadoran Civil War (1979-1992). The three scatterplots are similar in terms of scagnostics: high Monotonic and high Outlying. Looking at these distributions, one might wonder if there are other years for which that situation happened again or if there were other economic/social factors that also were affected by the wars in the same years. Inspecting scagnostics features, we can easily identify similar situations (similar scatterplots). This is especially important in a real-time environment when a lot of data come in batches, such as credit card fraud.

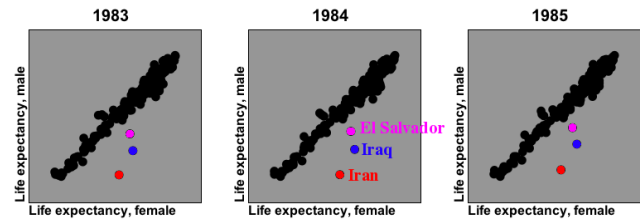


Figure 4: Three scatterplots of life expectancy of male vs. female in 1982, 1983, and 1984.

Answer 2: Having the similarity measure, we can group similar scatterplots and then create an abstract picture of the huge scatterplot collection. This is also the main purpose of using the similarity measure in this paper.

The dissimilarity of two scatterplots is computed as the squared Euclidean distance in feature space. The most obvious benefit of our parameterization is to reduce the complexity of comparing two scatterplots from  $O(n)$  to  $O(1)$  where  $n$  is the number of data points. That is, if we can characterize a scatterplot with 9 scagnostic measures (monotonicity, clumpiness, etc.), then we can make

Table 1: Characteristics of datasets used for testings and demonstrations in the following sections.

Datasets	# Instances	# Attributes	# Scatterplots
Breast Cancer	569	32	496
University ranking	127	33	528
US employment	144	44	946
Sonar	208	60	1,770
US money	64	79	3,081
Libras	360	91	4,095
Subway	423	104	5,356
Communities	1,994	128	8,128
Madelon	1,042	500	124,750
Arcene	900	3,000	4,498,500

comparisons directly on these measures (instead of point to point comparisons). The tradeoff here is, of course, that we might lose details of scatterplots.

We confine our model at this point to 2D scatterplots. There is nothing preventing us from computing most scagnostics in higher dimensions, but display issues come into play as the dimensionality increases. We believe that analysts are more familiar with 2D scatterplots than with more exotic displays, but that is a belief that requires testing in the future.

### 3.2 Datasets

We will review the issues involved in the ScagExplorer testbed mainly through examples. We use datasets retrieved from the UCI Repository [7] and other sources to demonstrate the performance of ScagExplorer. Table 1 summarizes prominent aspects of these datasets ordered by the number of attributes.

The US employment data comprise monthly statistics on various aspects of the US economy over 12 years from 2000 to 2011. There are 44 variables represented in the dataset: Employment Rate of major economy sectors such as Construction, Manufacturing, Financial Activities, etc. Data and variable descriptions can be found at <http://www.bls.gov/data/>. There are 144 data points in each scatterplot (144 months of 12 years) to examine.

The National Research Council (NRC) ranking data comprise university rankings in Mathematics in 2006. There are 33 variables represented in the dataset: R-Rankings, S-Rankings, ranking factors and information on 127 universities in the US. For S-Rankings, programs are ranked highly if they are strong in the criteria that scholars say are most important. For R-Rankings, programs are ranked highly if they have similar features to programs viewed by faculty as top-notch. Overall, we have 528 scatterplots with 127 data points (127 universities) to examine.

## 4 SCAGEXPLORER COMPONENTS

This section explains our approach in detail. Figure 5 shows a schematic overview:

1. **Processing:** Our approach computes nine scagnostics measures of each scatterplot in the input SPLOM. Then, scatterplots are clustered based on scagnostics space. At the end of Processing, we have a list of leader plots and a collection of followers (or children) for each leader.
2. **Visualization:** The leader plots in each cluster are displayed in the force-directed layout.
3. **Interaction:** Users can select a leader to see all similar plots in that cluster or filter scatterplots by their features.

Information visualization systems should allow one to perform analysis tasks that largely capture people’s activities while employ-

ing information visualization tools for understanding data [3]. The ScagExplorer implements four basic analysis tasks:

- **Clustering:** group scatterplots using their scagnostics measures (see Section 4.1 and Section 4.2).
- **Brushing:** select a leader plot to expand all similar plots in that cluster (see Section 4.3).
- **Sorting:** sort scatterplots based on their relevance to the leader scatterplot (see Section 4.3).
- **Filtering:** find scatterplots satisfying filtering conditions on their scagnostics (see Section 4.4).

### 4.1 Clustering Algorithm

The scatterplot matrix is a useful tool for displaying the correlation between a pair of variables. However, it is easy to run out of space as the number of variables increases. There are several solutions to deal with this scalability problem:

- **Dimension reduction:** preselect a subset of dimensions. The advantage of this approach is that we downsize the SPLOM. However, information loss is a disadvantage of this approach.
- **An alternative strategy is lensing [8].** However, to obtain an overview of the whole data, this approach is time consuming.
- **SPLOM reordering approaches:** Wilkinson et al.[27] sort the variables in the raw data SPLOM using the size of the loadings on the first principal component of the scagnostic measures. However, features sorting suggests clusters but it is not a clustering procedure. In other words, it does not guarantee that the two adjacent plots are similar. Lehmann et al.[19] used a heuristic optimization algorithm to reorder dimensions based on a scagnostics measure. This reordering method concentrates on the best plots in different regions, based on the similarity of the dimensions. Therefore, the number of relevant regions is arbitrary. Due to these limitations, we allow the scatterplots to break out of the SPLOM to form clusters.

We use the leader algorithm [14] to cluster scatterplots. The inputs to this algorithm are the list of all scatterplots and an initial threshold  $r$  (radius around a cluster’s center). Here is the summary of the algorithm:

1. We initialize the leader list  $L = \emptyset$ .
2. For each scatterplot  $S_i$ , we find the nearest leader (center) in  $L$  which has the squared Euclidean distance to  $S_i \leq r$ .
3. If we could not find a nearest leader satisfying this condition, we make  $S_i$  a new leader and add  $S_i$  into  $L$ .
4. Otherwise, we add  $S_i$  to the follower list of the nearest leader.
5. Repeat steps 2-4 for all scatterplots  $S_i$ .
6. Now we have a complete leader list  $L$ , we repeat step 2 and step 4 once for all scatterplots  $S_i$  to avoid the mistakes of finding the nearest leader when  $L$  is incomplete. We do not need to repeat step 3 because we are simply reassigning followers in case closer leaders emerged in the first pass through the data. This reassignment is similar to the iterative reassignments in the k-means algorithm [14], but we do only one more pass through the data. Thus, the computational complexity of the leader algorithm is considerably less than that for k-means, as [14] demonstrates.

Why not directly use many other well-known clustering algorithms like K-Means? First, the leader algorithm is rather a pre-processing step producing a small number of leader scatterplots that can represent a huge collection of scatterplots. Therefore, it is not sensitive to the actual number of clusters in the data. Second, the complexity of the leader algorithm is  $O(p)$  (where  $p$  is the number of scatterplots). This means in principle that the leader algorithm can handle higher-dimension datasets. In contrast, other

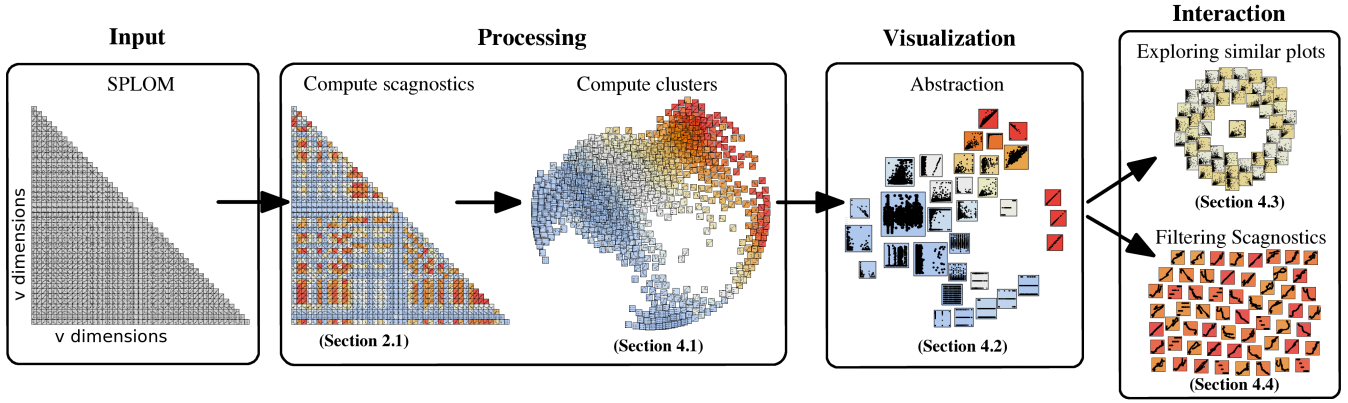


Figure 5: Schematic overview of ScagExplorer.

well-known clustering algorithms require at least polynomial time complexity.

Too many leaders makes the visualization too busy while too few leaders over-summarizes the data set. We therefore limit the size of the leader list  $L$  from  $\log_2 p$  to  $2\log_2 p$  to avoid these problems. For a smaller dataset of 50 dimensions, we expect from 10 to 20 leader plots. For a larger dataset of 1,000 dimensions, we expect from 18 to 36 leader plots. This means that we might need to adjust the threshold  $c$  and repeat steps 1-6 a few times to get the right number of leaders.

How to select the initial threshold  $r$ ? The distribution of squared Euclidean distances is approximately a noncentral Chi-square variable, assuming the distances themselves are roughly normal [16]. However, we chose an empirical method on real data in order to relax the normality assumption. From results on 20 different datasets of various sizes retrieved from the UCI repository [7], the threshold  $r$  that produced approximately 20 clusters varied from 0.5 (for small datasets with thousands of scatterplots) to 2.0 (for large datasets with millions of scatterplots) on a possible range of 0 to 9. Therefore, we initialize  $r = 2.0$ .

After each iteration, we check if the leader algorithm has produce the expected number of leaders. If not, we need to adjust  $r$  and repeat steps 1-6. *Binary search* is a quick way to get to the right threshold  $r$  for  $\log_2 p$  to  $2\log_2 p$  leaders. According to our experiments, *Binary search* can get to the right threshold  $r$  in fewer than 5 iterations on most of our test datasets.

## 4.2 Displaying the leader scatterplots

After having clusters and their leader plots, we now use the force-directed layout to place them on a 2D view. An alternative to this presentation is Multidimensional Scaling (MDS) [18] to project nine-dimensional scagnostics space to positions in 2D space. In the VaR technique [28], the glyph positions generated by MDS [18] are based on a distance matrix that records the correlation between each pair of dimensions in the dataset. However, it is impossible to get an optimal solution to this problem since there are many different correlation measures [17, 5]. Another drawback of MDS is that it produces overlapped scatterplots for similar distance measures, which makes it difficult to visualize cluster sizes. In force-directed layout, every plot is a physical object which repels other overlapped plots.

The advantages of force-directed layouts are intuitiveness, flexibility, and interactivity. The main disadvantage is high running time. Since for every plot, we have to compute the attraction or repellant against all other plots, the running time at each iteration is  $O(l^2)$  (where  $l$  is the number of leader scatterplots). Since, we limit  $l$  in the range from  $\log_2 p$  to  $2\log_2 p$ , the running time at each

iteration is  $O(\log_2 p^2)$ .

In the force-directed layout, we first put all leader plots randomly in the output panel and we then allow them to interact to find relevant leader plots based on their scagnostics measures. Consequently, relevant leaders are grouped together based on their scagnostics features. This makes it easier to interpret the clustering results. Here is the summary of the force-directed algorithm:

1. For each pair of leader scatterplots  $S_i$  and  $S_j$ , we compute a dissimilarity measure as the squared Euclidean distance in feature space.
2. We get the dissimilarity cut  $D$  initialized as  $r + 0.2$  where  $r$  is the radius of the clustering algorithm. We then define  $D_{ij} = \text{Dissimilarity}(S_i, S_j) - D$ .
3. We compute  $\vec{U}_{ij}$  as the unit vector from  $S_i$  to  $S_j$ .
4. If  $D_{ij} \leq 0$ ,  $\vec{F}_{ij}$  is the attraction between  $S_i$  and  $S_j$  computed by the following equation:

$$\vec{F}_{ij} = D_{ij} * \vec{U}_{ij} \quad (1)$$

5. If  $D_{ij} > 0$ ,  $\vec{F}_{ij}$  is the repulsion of  $S_j$  on  $S_i$ .

$$\vec{F}_{ij} = \frac{D_{ij} * \vec{U}_{ij}}{\text{Distance}(S_i, S_j)} \quad (2)$$

6. The force applied on  $S_i$  is the sum of forces by all scatterplots on  $S_i$  ( $l$  is the number of leader scatterplots):

$$\vec{F}_i = \sum_{i=1}^l \vec{F}_{ij} \quad (3)$$

7. Repeat steps 3-6 for all leader scatterplots  $S_i$ .

The algorithm can be stopped manually when users feel happy with the configuration or automatically when there is no more improvement (all similar leaders are close to each other). User can also increase  $D$  to form fewer but larger clusters and vice versa. Notice that in Equation 1, the attraction between  $S_i$  on  $S_j$  does not depend on their distance. This assures that similar plots can come close to each other no matter where they are in the display.

Figure 6 shows how we display the leader plots of four different datasets in the forced-directed layout. In particular, each frame summarizes thousands of scatterplots in each dataset. The datasets are Sonar (60 attributes or 1,770 scatterplots), Communities (128 attributes or 8,128 scatterplots), Madelon (500 attributes or 124,750 scatterplots), and Arcene (3,000 attributes or 4,498,500 scatterplots). The size of each leader plot is computed based on

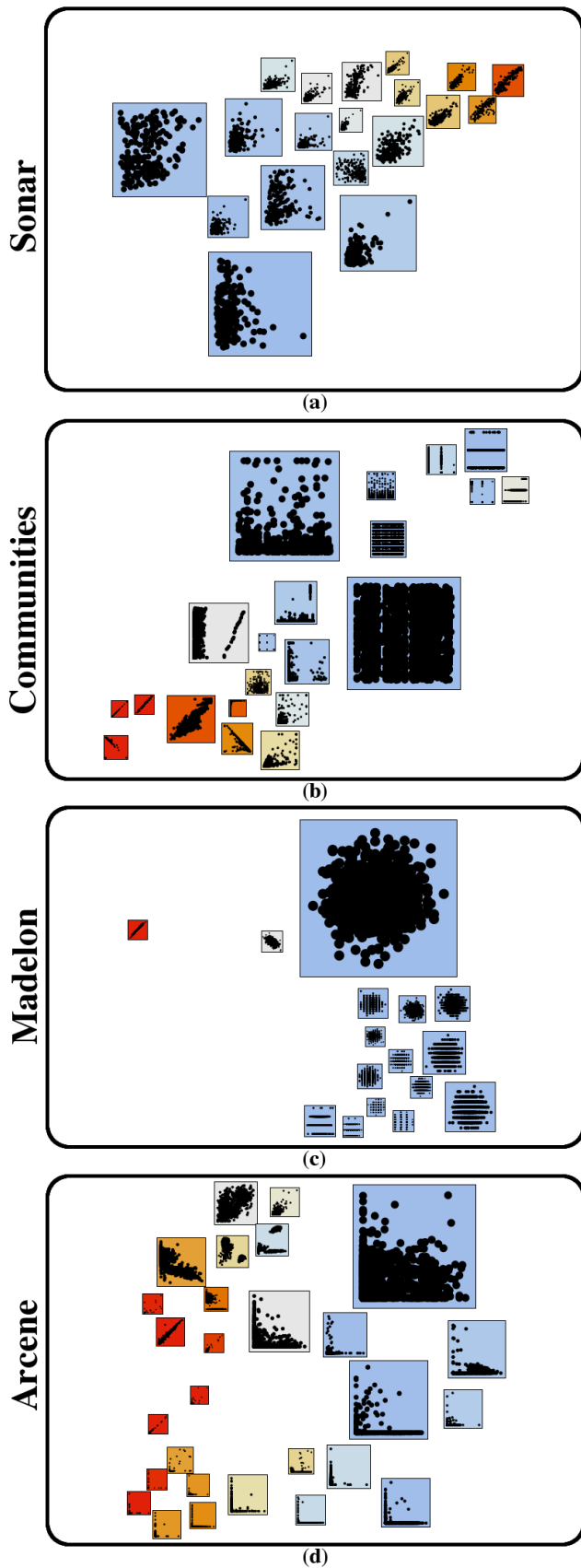


Figure 6: Visualizing the leader scatterplots in the Sonar, Communities, Madelon, and Arcene data.

its cluster size (or the number of scatterplots in each cluster). The Kelvin temperature color scale is adopted to highlight Monotonicity (red plots are high Monotonic, blue plots are low Monotonic).

Users can select a different scagnostics measure from a list box to highlight and/or align leader plots. In Figure 7, we aligned the leader plots on X-axis based on scagnostics measures. Unlike the layout in Figure 6 grouping plots based on all nine scagnostics measures, this layout groups plots based on only one measure (the selected measure). This alignment reveals the density distribution of scatterplots on the selected measure. In this alignment, the leader plots are pulled to their scagnostics locations on the X-axis (the plots might be pulled a bit to the left or right of its intended location due to the collisions with other plots sharing similar values on the selected measure). The Y-axis shows the concentration of scatterplots at the specific values on the X-axis. In Figure 7, we aligned the leader plots of three different datasets on Striated, Stringy, and Outlying respectively. The datasets are Breast Cancer (32 attributes or 496 scatterplots), US employment (44 attributes or 946 scatterplots), and Subway (104 attributes or 5,356 scatterplots). Notice that we have requested to display circles instead of rectangles to present scatterplots in Figure 7(b). This option creates a better effect on displaying density compared to rectangle shapes. In Figure 7(c), we have requested to display the number of scatterplots in each cluster on top of each leader plot.

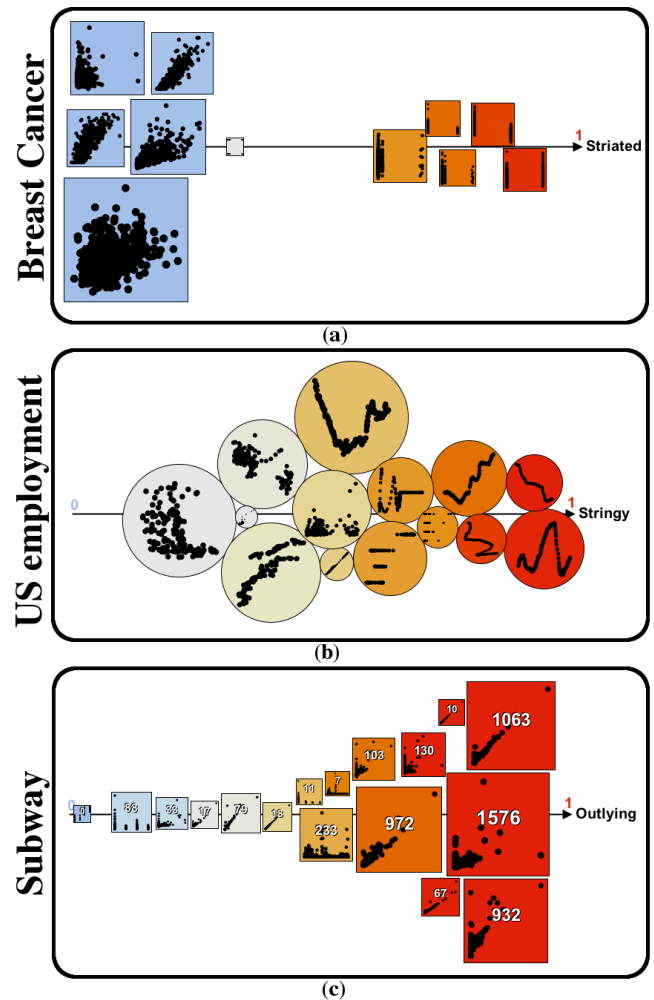


Figure 7: Visualizing the Breast Cancer, US employment, and Subway data. The leader plots in each dataset are colored and aligned by Striated, Stringy, and Outlying respectively.

Why not use a histogram for showing the density distributions of a particular scagnostic measure? As depicted in Figure 7(c), the Subway data are very highly Outlying. A histogram showing the density distributions of Outlying is a quick way to come up with the same conclusion. However, there are many configurations of data points in a scatterplot containing outliers. Users can not navigate through more than five thousand scatterplots to understand all Outlying configurations in the Subway data. ScagExplorer helps users to achieve this task by embedding the typical scatterplots (the leader plots with different sizes showing their popularities) into the density distribution graph. Moreover, it takes only 0.1 second to run the leader algorithm in this case. Since the leader algorithm is linear to the number of scatterplots, it is as fast as the binning process of histograms.

Users can now select from a list box to see the density distributions on different measures. It takes a few seconds for the transitions. The forced-directed layout makes transitions between different alignments smoothly. We only have to tell the leaders where they should go, then the leader plots fit themselves into the display area and avoid overlappings between them.

### 4.3 Exploring similar scatterplots

After having an overall idea of all scatterplots in the data, one may want to request the details in each cluster. This can be done by a simple click on a leader plot. Figure 8(a) shows an example for the Libras data. In particular, scatterplots are colored by their Monotonicity. The selected leader is in the center surrounded by scatterplots in its cluster. On the right, we link where they are in SPLOM. The same effect is shown in 8(b) with a different leader. The linked SPLOMs reveals the monotonic pattern: the closer to the diagonal, the higher Monotonicity. This is because variables in input data have been ordered so that highly correlated variables locate close to each other.

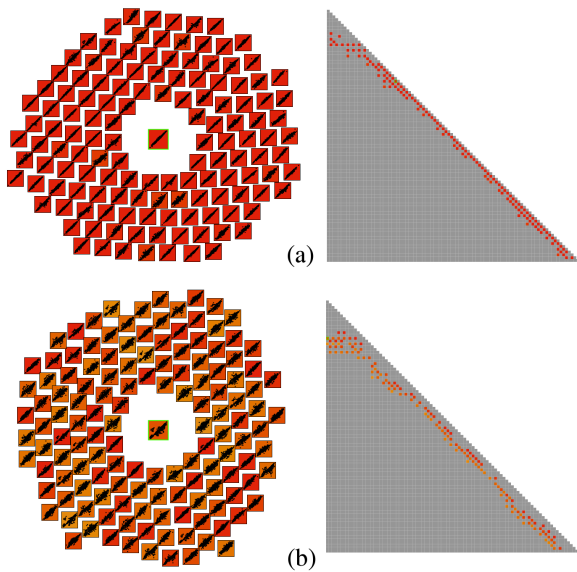


Figure 8: Expanding all scatterplots of a cluster in the Libras data.

### 4.4 Filtering Scagnostics

Another way to drill-down the large collection of scatterplots is filtering by scagnostics measures. Unlike the previous sections, we work (filter) directly on every scatterplot in the data (not the leader plots) in this section. In addition, we don't use the force-directed layout here. Figure 9(a) shows an example of filtering scagnostics by parallel coordinates. The data are NRC university rankings in

Mathematics. Each coordinate corresponds to a scagnostic measure. Colors are used to differentiate scagnostics. All scagnostics are in a common range from 0 (left) to 1 (right). There are 528 scatterplots in the NRC university ranking data, and thus the same number of polylines in parallel coordinates. The symmetric graph on each coordinate shows the density distribution of scatterplots in the entire search space according to each measure. When we are filtering a measure, ScagExplorer updates the density graphs on other measures showing only the remaining plots satisfying the filtering conditions. This guides users on making interactive scagnostic selections. Figure 9(b) shows another example of the US money data. The density distribution of scatterplots indicates that the data have very high Skewness and Monotonicity.

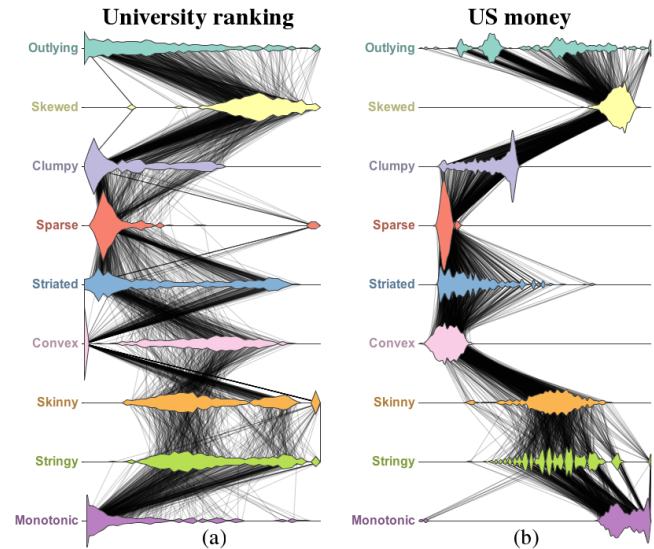


Figure 9: Density distribution of scatterplots by each scagnostic.

Figure 10(a) shows an example when this filter is applied on Monotonicity ( $\text{Monotonicity} \geq 0.5$ ). When we filter plots on one scagnostic, other graphs showing the distribution of selected scatterplots are updated. The right frame shows 31 plots satisfying the condition. We can obtain correlated variables from these high Monotonic scatterplots. After viewing pairs of variables that are highly correlated, one can request to see the variable relationship graph. Figure 10(b) shows the variable relationship graph of 31 monotonic plots in Figure 10(a). Each node in this graph represents a variable and each edge exists if a pair of variables exists in Figure 10(a). Red edges connect highly correlated variables. Notice that Research Activity is the main variable involved with both R Rankings and S Rankings. Other variables which have moderate influence on rankings are percent of faculty with grants, citation per publication, average number of PhD Students graduated, and publication per allocated faculty.

## 5 EVALUATION

ScagExplorer is not meant to be an end-user application, but rather a testbed for evaluating the feasibility of this approach. Consequently, we are not prepared at this time to conduct a comprehensive user study that employs analysts investigating real datasets. By contrast, we focus here on evaluating the performance of these algorithms. We need to assess 1) whether the computations are practical on large datasets, particularly with regard to running times, 2) whether the leader algorithm yields suitable exemplars, and 3) whether the force-directed clustering can produce clear clusters on well-structured test data.

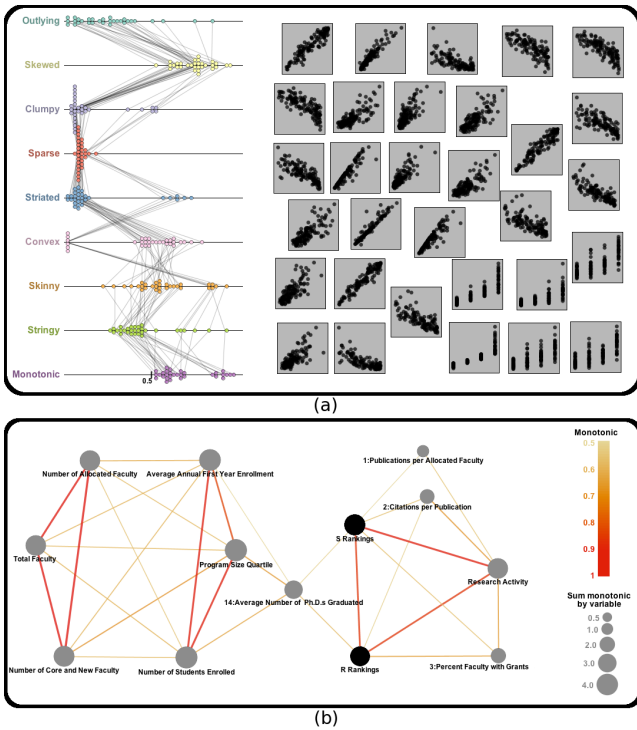


Figure 10: Filtering high Monotonic scatterplots (Monotonicity  $\geq 0.5$ ) in the NRC university ranking data.

### 5.1 Running Times

We investigated the performance of our ScagExplorer testbed on huge data in terms of  $n$  (number of observations) and  $p$  (number of scatterplots). All tests were performed on a 2.3 GHz Intel Core i5, Mac OS X Version 10.7.5, 4 GB RAM running Java 1.6 and Processing 1.5.1. The graphs in Figure 11 show computation time broken down into the time to bin the  $n$  data points (observations), compute scagnostics, run the leader algorithm, and organize the leader scatterplots in the force-directed layout. Here are some observations from empirical analysis:

- Running the leader algorithm and organizing the force-directed layout do not depend on  $n$  since they work on scagnostics space. Moreover, computing scagnostics is almost independent of  $n$  since proximity graphs are generated on binned data. Only the stage of binning the data points is linearly dependent on  $n$  (more details can be found [26]).
- The bottleneck of our approach is at the stage of computing scagnostics. However, this stage is completely parallelizable. An alternative to ameliorate this problem is to sample the input data.
- The complexity of the leader algorithm is  $O(p)$  and the complexity of the force-directed clustering is  $O(\log_2 p^2)$ .

Overall, computation time of the ScagExplorer is roughly  $O(np)$ .

### 5.2 The Leader Algorithm

We next tested our algorithms using a Monte Carlo structured dataset. In particular, we generated nine groups of 50 scatterplots with high values on nine scagnostics measures (the scatterplots on the right of Figure 2). The scatterplots in each group were perturbed with a uniform $[-.1, .1]$  random variable. The leader algorithm correctly generates nine clusters of 50 scatterplots. Figure 12 shows Outlying, Skinny, and Stringy clusters. Leader plots are in the centers of each cluster.

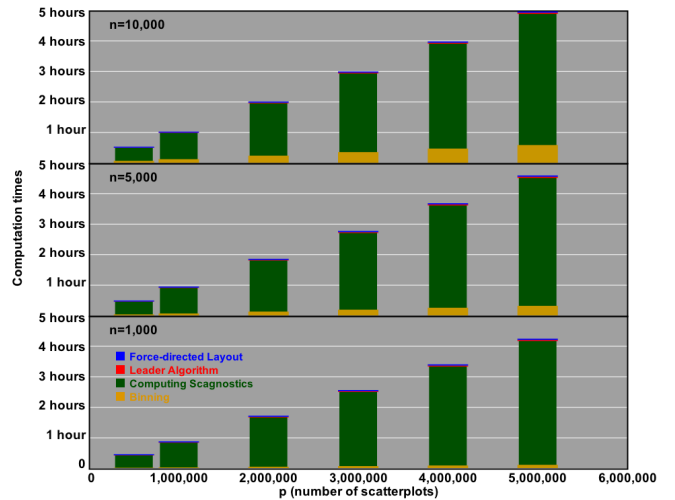


Figure 11: Computation times (in hours) for large datasets where  $n$  is the number of observations and  $p$  is the number of scatterplots.

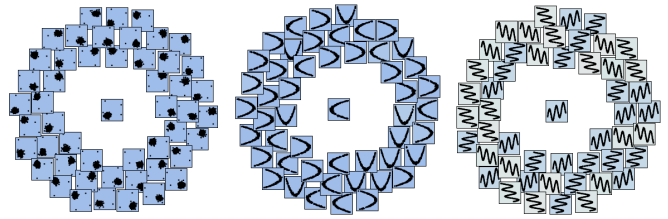


Figure 12: Three clusters in the Monte Carlo test dataset.

### 5.3 The Force-directed Layout

The second test on the Monte Carlo test dataset is whether the force-directed layout can produce clear clusters on well-structured test data. Figure 13 shows that the force-directed layout correctly groups similar scatterplots into the same clusters and separates the clusters based on their inter-cluster dissimilarities.

## 6 CONCLUSIONS

The performance of our ScagExplorer testbed on real datasets and on the Monte Carlo test datasets suggest that the algorithms outlined in the paper could be used to embed scagnostics analytics in platforms designed to handle high-dimensional datasets.

ScagExplorer highlights the usefulness of examining raw data distributions rather than reductive statistical summaries or aggregations (as in OLAP cubes). As we have seen in the real examples, pairwise displays are rarely Gaussian or otherwise well-behaved. Furthermore, ScagExplorer makes it possible to examine pairwise scatterplots in a coherent framework. While parallel coordinates and other multivariate displays (glyphs, projections, profiles, etc.) have their uses, it is helpful to have a tool that allows a user to explore data in the familiar form of the scatterplot.

We note that the time for comparing two scatterplots is  $O(1)$  compared to  $O(n)$  because we are searching on nine scagnostics, not on individual points (where  $n$  is the number of data points). Therefore, the benefit of our approach is the enormous compression we achieve by collapsing similarity searches from  $O(pn)$  to  $O(p)$  through the use of scagnostics. Moreover, we implemented a quick clustering algorithm [14] with the complexity  $O(p)$  or  $O(v^2)$  compared to the greedy variable reordering algorithm [19] which requires  $O(v^3)$  (where  $v$  is the number of variables in the data). This

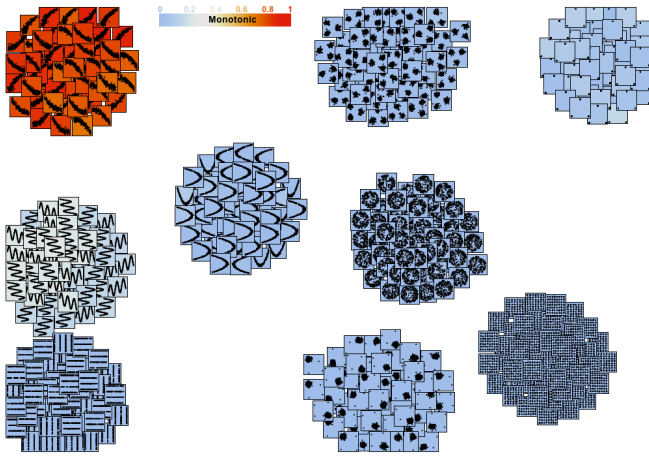


Figure 13: Force-directed layout produces nine clusters in the Monte Carlo test dataset. Scatterplots are colored by their Monotonicity.

provides ScagExplorer with the scalability to handle huge datasets up to thousands of dimensions.

But there is also a drawback of the approach: one problem of the forced-directed layout is that random initial layout would yield different clusters of coherent plots. Therefore, different runs end up with different configurations. However, the final configurations of the same data are consistent because relevant leaders are grouped together. This provides a comprehensive summary of the input data. An alternative to ameliorate this problem is to use MDS to assign the initial positions for the leader plots. MDS helps to bring similar leaders together in the initial layout, and so the final configurations are not so different on different runs.

What does this mean for visual analysis? This paper proposes a testbed for visualizing high dimensional data where the number of scatterplots is too large to be visualized by an ordinary SPLOM. The greater runtime efficiency allows ScagExplorer to provide a quick and comprehensive summary of the input data. Then, we can drill-down in the data by inspecting a cluster or filtering using a target scagnostics measure.

## ACKNOWLEDGEMENTS

This work was supported by NSF/DHS grant DMS-FODAVA-0808860.

## REFERENCES

- [1] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the 1998 ACM SIGMOD international conference on Management of data, SIGMOD '98*, pages 94–105, New York, NY, USA, 1998. ACM.
- [2] G. Albuquerque, M. Eisemann, and M. Magnor. Perception-based visual quality measures. In *IEEE VAST*, pages 13–20, 2011.
- [3] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *Proc. of the IEEE Symposium on Information Visualization*, pages 15–24, 2005.
- [4] A. Anand, L. Wilkinson, and T. N. Dang. Visual pattern discovery using random projections. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pages 43–52, 2012.
- [5] M. Ankerst, S. Berchtold, and D. A. Keim. Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In *Proceedings of the 1998 IEEE Symposium on Information Visualization, INFOVIS '98*, pages 52–, Washington, DC, USA, 1998. IEEE Computer Society.
- [6] I. Assent, R. Krieger, E. Müller, and T. Seidl. Visa: visual subspace clustering analysis. *SIGKDD Explor. Newsl.*, 9(2):5–12, Dec. 2007.

- [7] A. Asuncion and D. Newman. UCI machine learning repository, 2007.
- [8] T. N. Dang, A. Anand, and L. Wilkinson. Timeseer: Scagnostics for high-dimensional time series. *Visualization and Computer Graphics, IEEE Transactions on*, 19(3):470–483, 2013.
- [9] A. Dasgupta and R. Kosara. Pargnostics: Screen-space metrics for parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 16:1017–2626, 2010.
- [10] H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29:551–559, 1983.
- [11] L. Fu. Implementation of three-dimensional scagnostics. Master's thesis, University of Waterloo, Department of Mathematics, 2009.
- [12] M. Gregory and B. Shneiderman. Shape identification in temporal data sets. In J. Dill, R. Earnshaw, D. Kasik, J. Vince, and P. C. Wong, editors, *Expanding the Frontiers of Visual Analytics and Visualization*, pages 305–321. Springer London, 2012.
- [13] S. Guha, P. Kidwell, R. Hafen, and W. S. Cleveland. Visualization databases for the analysis of large complex datasets. *Journal of Machine Learning Research - Proceedings Track*, 5:193–200, 2009.
- [14] J. Hartigan. *Clustering Algorithms*. John Wiley & Sons, New York, 1975.
- [15] H. Hochheiser and B. Shneiderman. Dynamic query tools for time series data sets: Timebox widgets for interactive exploration. *Information Visualization*, 3:1–18, March 2004.
- [16] N. L. Johnson, S. Kotz, and N. Balakrishnan. *Continuous Univariate Distributions, Vol. 1 (Wiley Series in Probability and Statistics)*. Wiley-Interscience, second edition, 1994.
- [17] R. A. Johnson and D. W. Wichern, editors. *Applied multivariate statistical analysis*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.
- [18] J. B. Kruskal and M. Wish. Multidimensional scaling, 1978.
- [19] D. J. Lehmann, G. Albuquerque, M. Eisemann, M. Magnor, and H. Theisel. Selecting coherent and relevant plots in large scatterplot matrices. *Comp. Graph. Forum*, 31(6):1895–1908, Sept. 2012.
- [20] L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: a review. *SIGKDD Explor. Newsl.*, 6(1):90–105, June 2004.
- [21] J. Schneidewind, M. Sips, and D. Keim. Pixnostics: Towards measuring the value of visualization. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 199–206, Baltimore, MD, 2006.
- [22] J. Seo and B. Shneiderman. A rank-by-feature framework for interactive exploration of multidimensional data. *Information Visualization*, 4(2):96–113, July 2005.
- [23] M. Sips, B. Neubert, and J. Lewis. Selecting good views of high-dimensional data using class consistency. *Computer Graphics Forum*, 28:831–838, 2009.
- [24] A. Tatu, F. Maa, I. Frber, E. Bertini, T. Schreck, T. Seidl, and D. A. Keim. Subspace Search and Visualization to Make Sense of Alternative Clusterings in High-Dimensional Data. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 63–72. IEEE CS Press, 2012.
- [25] A. Tatu, L. Zhang, E. Bertini, T. Schreck, D. A. Keim, S. Bremm, and T. von Landesberger. ClustNails: Visual Analysis of Subspace Clusters. *Tsinghua Science and Technology, Special Issue on Visualization and Computer Graphics*, 17(4):419–428, Aug. 2012.
- [26] L. Wilkinson, A. Anand, and R. Grossman. Graph-theoretic scagnostics. In *Proceedings of the IEEE Information Visualization 2005*, pages 157–164. IEEE Computer Society Press, 2005.
- [27] L. Wilkinson, A. Anand, and R. Grossman. High-dimensional visual analytics: Interactive exploration guided by pairwise views of point distributions. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1363–1372, 2006.
- [28] J. Yang, A. Patro, S. Huang, N. Mehta, M. O. Ward, and E. A. Rundensteiner. Value and relation display for interactive exploration of high dimensional datasets. In *Proceedings of the IEEE Symposium on Information Visualization, INFOVIS '04*, pages 73–80, Washington, DC, USA, 2004. IEEE Computer Society.