Visual Pattern Discovery using Random Projections

Anushka Anand* Leland Wilkinson† Tuan Nhon Dang‡

Department of Computer Science, University of Illinois at Chicago

ABSTRACT

An essential element of exploratory data analysis is the use of revealing low-dimensional projections of high-dimensional data. Projection Pursuit has been an effective method for finding interesting low-dimensional projections of multidimensional spaces by optimizing a score function called a projection pursuit index. However, the technique is not scalable to high-dimensional spaces. Here, we introduce a novel method for discovering noteworthy views of high-dimensional data spaces by using binning and random projections. We define score functions, akin to projection pursuit indices, that characterize visual patterns of the low-dimensional projections that constitute feature subspaces. We also describe an analytic, multivariate visualization platform based on this algorithm that is scalable to extremely large problems.

Keywords: Random Projections, High-dimensional Data.

Index Terms: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)— [H.2.8]: Database Applications—Data Mining

1 INTRODUCTION

This work is motivated by the lack of pattern detection methods that are efficient in high-dimensional spaces. While many of these methods [16, 10, 35, 28] are effective in low-dimensional spaces, their computational complexity prevents their application in very high-dimensional spaces. While there are approaches to speed up these methods [24, 49, 14], most involve parallelization schemes, computationally complex ideas or ad-hoc pre-processing methods.

By contrast, we have attempted to integrate a recent method called random projections, in an iterative framework, to detect visual patterns in high-dimensional spaces. The main use of the random projection theorem in data mining has been as a preprocessing step in order to reduce the dimensionality of a space to a manageable subspace before applying classification algorithms such as Support Vector Machines or decision trees that are not readily scalable to very high-dimensional spaces. CHIRP [46] proposed a different approach: it is a visual-analytic classifier, based on iterated random projections, that approximates the distribution-free flexibility of Projection Pursuit without the computational complexity. This motivated us to see whether guided random projections could be used to construct a visual analytics platform for discerning meaningful structures in very large datasets. In this paper we investigate such an approach to find dimensions that contribute strongly to analytically meaningful features. Such features capture visual patterns of point sets that drive substantive interpretations. The strength of our method is in exploiting *non-axis-parallel* subspaces (i.e. non-orthogonal projections).

Making this effective requires us to analyze how to visually explore these random projections, as models based on subspaces, that together characterize a score. Because we require a large number of random projections to find substructures in high-dimensional spaces, visualizing these collections of random projections to understand views of the data is challenging. The Subspace Explorer we developed incorporates multiple multivariate perspectives of the subspace with extensive interaction and drill-down capabilities to understand the contribution of variables.

Our contributions in this paper are:

- An algorithmic scheme, based on random projections, to find low-dimensional views of interesting visual substructures in very high-dimensional data.
- A methodology to compute features in high-dimensional space based on prior work in 2-D [47].
- An Exploratory Data Analysis (EDA) tool, the Subspace Explorer, that can be used to explore multiple perspectives of low-dimensional projections.

We describe related work in the following section. Then, we describe our approach and the Subspace Explorer and finally we describe our validation tests and examples of use with real data before ending with concluding thoughts.

2 RELATED WORK

Our visual analytic application depends on several areas of prior research. The first concerns *projections*, which reduce dimensionality in vector spaces in order to reveal structure. The second involves *manifolds*, which use graph-theoretic concepts to reveal structure in point distributions. The third concerns *visual features*, which can describe structures in multivariate data.

2.1 Projections

To discover low-dimensional projections clearly discriminating clusters, researchers have applied Multidimensional Scaling (MDS) [33] and Principle Component Analysis (PCA) [12]. These are computationally expensive procedures when dealing with very high-dimensional data sets.

Projection pursuit has been combined with the Grand Tour [6, 13], which smoothly animates a sequence of projections, resulting in a dynamic tool for EDA. Projection Pursuit has been used to find class-separating dimensions [29] and combined with tours to investigate Support Vector Machines [11]. These approaches allow one to explore the data variables via projections that maximize class separability but can only handle a small number of variables (≤ 8 for clear tour displays).

Axis-parallel 2-D projections of data are ranked on a class separability measure [41] for multivariate visual exploration and used with 1-D histograms to analyze variables and interactively construct a classifier [19]. However, using axis-parallel projections of data to find low-dimensional views with large class-separation [37] will be ineffective when there does not exist class separability in axis-parallel views of the data.

*e-mail: aanand2@uic.edu

†e-mail: leland.wilkinson@systat.com

‡E-mail: tdang@cs.uic.edu

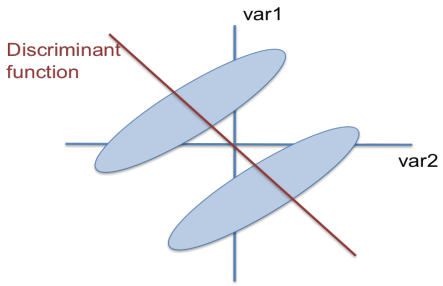

Figure 1: Fisher's linear discriminant separating two groups.

2.1.1 The Problem with Axis-parallel Projections

There are two serious problems with axis-parallel approaches: 1) computations on pairs of dimensions grow quadratically with the number of dimensions, and 2) axis-parallel projections do a poor job of capturing high-dimensional local geometry. While the pursuit of axis-parallel projections can be interesting and may well work for some smaller data sets, there is no reason to expect that they can reveal interesting structures in data. Moreover, the published examples of "high-dimensional" problems involve relatively few dimensions (in the tens or hundreds). Our Cancer dataset example has over 16,000 dimensions. Dimensions of that magnitude have over a hundred million 2-D axis-parallel projections. Generating this number of plots and visually examining them is impractical.

Regarding the second problem, consider the following worst-case lemma:

> Let U be a set of $p(p-1)/2$ axis-parallel unit vectors in $\mathbb{R}^p$. Let x be a $p \times 1$ positive vector in $\mathbb{R}^p$ that represents a dimension of maximum interest (an interesting 1-D projection, a Bayes-optimal discriminating vector, a first principal component, ...). Let u_x be the element of U that is closest in angle to x. Then for any x, the maximum possible angle between u_x and x is $\theta_{max} = \arccos(1/\sqrt{p})$.

As p increases, θ_{max} heads toward 90 degrees rapidly (for $p = 20$, it's almost 80 degrees). In short, axis-parallel projections will tend to be non-informative for high-dimensional problems. This result is a corollary of the curse of dimensionality. Thus even for algorithms designed to search algorithmically through axis-parallel projections for optimal views, there is no guarantee that these can be found and no way of knowing that ones discovered are anywhere near optimal.

Further, we assert that algorithms that use margins alone to find multidimensional structure are based on mistaken assumptions. Figure 1 shows two clusters in 2-D space where neither variable alone is sufficient to determine the clusters. Fisher's linear discriminant is essentially one variable that is a linear combination of the two - the data projected onto the linear discriminant would clearly show two clusters. Some may argue that you get the same result by keeping both variables but in p-dimensional space you need all p variables and that is impractical when dealing with large-p datasets. This point is central to every argument subsequently in the paper. Any pattern recognition endeavor faces this problem; an example is subspace clustering [34]. Typical papers on subspace clustering demonstrate their value on well-separated, Gaussian or hyper-rectangle clusters in an embedding space [3, 7]. For such data, a directed search using greedy approaches can find these clusters. However, many real datasets present ill-posed problems [43]. Effective modeling requires regularization techniques. Or in our case, random projections.

2.1.2 Random Projections

Johnson-Lindenstrauss [26] proved that if a metric on $\mathbf{X} \in \mathbb{R}^p$ results from an embedding of $\mathbf{X}$ into Euclidean space, then $\mathbf{X}$ can be embedded in $\mathbb{R}^q$ with distortion less than $1+\varepsilon$, where $0 < \varepsilon < 1$, and $q = O(\varepsilon^2 log(|\mathbf{X}|))$. Remarkably, this embedding is achieved by projecting onto a random q-dimensional subspace such that, roughly speaking, the input points are projected onto spherically random hyperplanes through the origin. A random projection matrix $\mathbf{T}_{p\times q} \in \mathbb{R}^q$ is generated with identical independently-distributed entries with zero mean and constant variance. This is multiplied with the original matrix $\mathbf{X}_{n\times p} \in \mathbb{R}^p$ to obtain the projected matrix

$$\mathbf{B} = \frac{1}{\sqrt{q}}\mathbf{XT} \in \mathbb{R}^q$$

where $q << min(n, p)$. $\mathbf{B}$ preserves all pairwise Euclidean distances of $\mathbf{X}$ in expectation. This theorem has been applied with much success to many data mining approaches from classification to clustering [15, 9] primarily as a dimensionality reduction technique.

2.2 Manifolds

The manifold learning community [42, 8] has used nonlinear maps on k-Nearest Neighbor graphs to find informative low-dimensional representations of high-dimensional data. They assume that the data of interest lie on an embedded non-linear manifold within the higher-dimensional ambient space usually of much lower dimensionality. We are similarly interested in finding an embedded subspace but instead of using geodesic distances or neighborhood functions to approximate high-dimensional distances, we use random projections.

2.3 Features

In the Rank-by-Feature framework [22], all pairwise relations are ranked according to various statistical features such as correlation, uniformity, normality, etc. and users can explore 1-D or orthogonal 2-D views of variables using histograms, boxplots or scatterplots. Information-bearing 2-D projections of data in scatterplots are ranked on perceptually motivated goodness measures based on particular tasks [5]. Albuquerque et. al [4] describe quality metrics for multivariate visualization techniques like scatterplot matrices and parallel coordinates in relation to the related tasks of analyzing clusters and separating labeled classes. In these approaches 2-D projections are ranked and visually analyzed.

Turkay et al. [45] propose a model for dual analysis of items and dimensions through linked views of orthogonal projections of items and plots of summary statistics of the dimensions. The objective is to understand relationships between dimensions by brushing items. Their work is analogous to the Scagnostics [47] idea, as they characterize and visualize individual dimensions with summary statistics like the mean and skewness. In doing so, they transform data dimensions to feature space. By contrast, Scagnostics is closer to the spirit of Projection Pursuit indices than restrictive classical statistical summaries like means and skewness. Our work extends Scagnostics as we look at characterizing visual patterns within highly multivariate data and presenting them in an environment that guides exploration.

3 The Subspace Explorer

In this paper we describe a computationally efficient search algorithm for interesting views of very high-dimensional data using an approach based on random projections. To explore the subspaces described by the discovered random projections and get insight into the relationships between variables, we developed the Subspace Explorer which leverages multiple visualizations or views of multivariate data. To handle large datasets both in terms of n, the number of observations, and p the number of variables, we leverage binning and random projections as described in the following sections.

3.1 Binning

To efficiently deal with large datasets in our visual analytic platform, we first bin datasets with more than 2000 observations. We base the number of bins on a formula in [40]. Given n instances, we compute the marginal number of bins b using

$$b = c_1 \; log_2(n)$$

The constant c_1 is a parameter the analyst can manipulate depending on the level of compression required; larger values improve speed but reduce resolution. The application defaults to $c_1 = 100$, which is seen empirically (across a wide variety of datasets) to produce a sufficiently large number of bins to capture the data distribution. Multivariate data binning is usually performed by a k-Nearest Neighbor (kNN) grouping of the data or a segmentation of the kNN connectivity graph or a Voronoi tessellation or k-means clustering. We adapt a single run of the k-means algorithm to find b clusters with the modification that it be sensitive to datasets with outliers. First, we pick b data points at random as starting centroids, then for every data point we compute its distance to each b centroid and store the closest centroid and distance. The goal is to have a representative segmentation of the high-dimensional, many-points dataset, so outlier points or those far from most of the data points are placed into their own singleton bin. Those points whose distance to their closest centroid is greater than a threshold t are considered singleton bins, where

$$t = \hat{\mu} + c_2 \hat{\sigma}$$

Here $\hat{\mu}$ is the mean of the vector of closest distances and $\hat{\sigma}$ is the sample standard deviation of the same. The constant c_2 is a tunable parameter that is set to 5 based on empirical studies that ensure the balance between coverage and number of bins. Thus, the total number of bins may increase depending on the extra singletons discovered. Given the singleton bins, the other data points are assigned to their closest bin and each bin centroid is updated. This can be seen as a modified single run of the k-means algorithm with the resulting list of bins as the segmentation of the multivariate data space and the bin centroids as the new data points.

3.2 Selecting Random Projections

By working with point-to-point distances in low-dimensional, *non-axis-parallel* subspaces, we escape the curse of dimensionality following the successful results by researchers in the manifold learning community [18]. The original Johnson-Lindenstrauss proof involved Gaussian weights so the input matrix $\mathbf{X}$ is multiplied by a dense matrix of real numbers. This is a computationally expensive task for large data. Achlioptas [1] replaced those projections with simpler, faster operations as he showed that using unit-weighted projections do not jeopardize accuracy in approximating distances. The projection matrix he generated consists of identical independently-distributed random variables with values in $\{1, 0, -1\}$ drawn from a probability distribution, so

$$t_{ij} = \sqrt{s} \begin{cases} 1 \text{ with probability } \frac{1}{2s} \\ 0 \text{ with probability } 1 - \frac{1}{s} \\ -1 \text{ with probability } \frac{1}{2s} \end{cases}$$

He recommended using $s = 3$ which gives a three-fold speedup as only a third of all the variables are processed to construct the embedding. The key issue for all methods producing Johnson-Lindenstauss embeddings is that it must be shown that for any vector, the squared length of its projection is sharply concentrated around its expected value. Achlioptas' contribution is in determining the probability distribution for the random projection matrix such that for all vectors the concentration is as least as good as in the spherically symmetric case. He even shows a miniscule improvement while using such a spare projection matrix. The computation of the projection reduces to aggregate evaluation – summations and subtractions of randomly selected variables but no multiplications. Furthermore, Hastie et al. [31] showed that unit random weights for most purposes can be made *very* sparse. For better robustness, especially when dealing with heavy-tailed distributions, they recommend choosing $s = \sqrt{p}$. We use this sparse formulation to produce the projection matrix in our application.

Applying the Johnson-Lindenstrauss theorem yields a subspace whose inter-point distances are negligibly different from the same inter-point distances in very high-dimensional space. This enables the execution of relatively inefficient data mining methods without significant loss of accuracy. This also enables the fast generation of low-dimensional projections that can be evaluated and visualized. The subspaces generated from this framework differs from the subspaces analyzed in traditional subspace clustering approaches [34] in that each dimension of a q-dimensional (q-D) random projection (which can individually be considered a 1-D random projection) is a linear combination of the original data dimensions. Subspace clustering approaches consider a subset of the original data dimensions as the subspace to evaluate.

We generate a number, r, of random q-D projections as candidate subspaces and score each based on the visual pattern features we define. Then we present the 10 top-scoring q-D random projections in the Subspace Explorer. For our search, we generate $r = 50$ q-D projections to score. We set the default q for the q-D random projection space to be 20. The random projection dimension q is meant to be $O(log_2(n))$ from the theorem, but with the binning, this is a good estimate in most cases and it can be changed by the user of the platform easily, as needed. Generating one q-D random projection matrix is fast: for each 1-D random projection vector, we pick $s = \sqrt{p}$ variables and randomly assign $+/-1$ weights to them. Thus, picking redundant variables is less of a problem for random projections than it is for subset subspace methods. And redundancy is less likely to occur as p increases which is the focus of this research. We then use the projection matrix to get the q-D projected data that is scored by computing features as described below.

Our search strategy of generating and evaluating a random sample of r subspaces has its limitations. Increasing r or the size of the search space, increases the likelihood of finding the desired substructure, but, at the cost of time. Looking at 50 high-dimensional slices is a good compromise (which we determined empirically). Noise is another problem that plagues every data-mining algorithm. If there is a large amount of noise relative to the systematic elements of the problem, no data mining method, including Support Vector Machines and random projection based approaches, is going to perform well.

3.3 Computing Features

The most prevalent approaches to computing visual features in point sets in high-dimensional spaces have been based on 2-D axis-parallel projections [22, 5, 48]. Scagnostics [47] present a canonical set of nine features that have proven to be effective in moderate-dimensional problems. Not all of the Scagnostics features can be extended to higher-dimensions however. Here we describe those that we incorporate into The Subspace Explorer. These are the measures that are computed from the geometric Minimum Spanning Tree (MST) of the set of points [2].

Outlying: Tukey [44] introduced a non-parametric approach to outlier detection based on peeling the convex hull as a measure of the *depth* of a level set imposed on scattered points. We do not assume that the multivariate point sets are convex and want to detect outliers that may be located in interior, relatively empty regions. So, we consider a vertex whose adjacent edges in the MST all have

a weight (length) greater than ω to be an outlier. Setting ω using Tukey's gap statistic based on conventional boxplots produces a large number of outliers for large datasets ($n > 10,000$). So, w deviate from the original formula and adopt the robust methodology of Letter Value Boxplots [20] which get detailed information in the tails through order statistics at various depths going out from the median. We choose the formulation of ω, the depth after which points are labeled outliers, to be such that z percent of the sample size are considered outliers. This z can be varied by the analyst based on domain knowledge but defaults to 2%. The Outlying measure is the proportion of the total edge length due to edges connected to detected outliers.

$$c_{outlying} = length(T_{outliers})/length(T)$$

Clumpy: This Scagnostic is based on the Hartigan and Mohanty RUNT statistic [17]. The runt size of a *dendrogram* (the single-linkage hierarchical clustering tree) node is the smaller of the number of leaves of each of the two subtrees joined at that node. As there is an isomorphism between a single-linkage dendrogram and the MST, we can associate a runt size (r_j) with each edge (e_j) in the MST, as described by Stuetzle [39]. The clumpy measure emphasizes clusters with small intra-cluster distances relative to the length of their connecting edge and ignores runt clusters with relatively small runt size.

$$c_{clumpy} = \max_j \left[1 - \max_k \left[length(e_k) \right] / length(e_j) \right]$$

where j indexes edges in the MST and k indexes edges in each runt set derived from an edge indexed by j.

Sparse: Sparseness measures whether points are confined to a lattice or a small number of locations in the space. This can happen when dealing with categorical variables or when the number of points is extremely small. The 90th percentile of the distribution of edge lengths in the MST is an indicator of inter-point closeness.

$$c_{sparse} = q_{90}$$

Striated: Another common visual pattern is striation: parallel strips of points produced by the product of categorical and continuous variables. Let $V^{(2)} \subseteq V$ be the set of all vertices of degree 2 in V and let $I()$ be an indicator function. Striation counts the number of adjacent edges whose between-angle cosine is less than -0.75.

$$c_{striated} = \frac{1}{|V|} \sum_{v \in V^{(2)}} I(\cos\theta_{e(v,a)e(v,b)} < -0.75)$$

Stringy: With coherence in a set of points defined as the presence of relatively smooth paths in the MST, a stringy shape is determined by the count of degree-2 MST vertices relative to the overall number of vertices minus the number of single-degree vertices. Smooth algebraic functions, time series, points arranged in vector fields and curves fit this definition.

$$c_{stringy} = \frac{|V^{(2)}|}{|V| - |V^{(1)}|}$$

3.4 Displaying Subspaces

The Subspace Explorer is an EDA dashboard that presents charts with different views of the data and facilitates multiscale exploration of data subspaces through rich interactivity in a manner similar to iPCA [23] which facilitates understanding the results of PCA. Others [27] have noted that connecting multiple visualizations by linking and brushing interactions provides more information than considering the component visualizations independently. We combine several geometric techniques - SPLOMs, parallel coordinate plots and biplots - along with a graph-based technique and a radar plot icon to visualize collections of random projections and the subspaces of high-dimensional data they describe as seen in Figure 2.[1].

The overall interface design of the Subspace Explorer is based on multiple coordinated views. Each of the six data views represents a specific aspect of the input data either in dimension subspace or projection space, and are coordinated in such a way that any interaction with one view is immediately reflected in all the other views allowing the user to infer the relationships between random projections and dimension subspaces. Data points are colored according to class labels if they exist or in a smoothly increasing gradient of colors from yellow to purple based on the order of the data points in the input file. The smooth gradient indicates the sequence of observations which may be of interest when looking at Stringy shapes that point to time series data.

Because it is designed for expert analysts, Subspace Explorer trades ease of first use for analytic sophistication. Consequently users require some training to be familiarized with the definition of random projection subspaces and to read and interact with the various plots. The objective of the Explorer is to aid in the interpretation of random projection subspaces in terms of the original data variables. This is complicated by the fact that each dimension of the random projection subspace is a linear combination of a subset of the original variables. Therefore the user can switch between looking at the projected space (the q-D random projection matrix B) in the Projection Space Views and the data subspace (the subset of dimensions used in the random projections) in the Data Space Views.

The Control options allow the user to select an input data file, specify whether the data file has a class label as the last column (this information is used to color the data points in the visualization) and a row label as the first column, the score function to maximize, a random number seed (for the random number generator) and the dimensionality, q, of the random projection space (the default is set to 20 as described in Section 3.2). The Update button causes the data file to be read, the data matrix to be binned, a number of q-D random projections to be generated and scored, the top 10 scoring projection scores to be shown in the Score View and the best scoring q-D projection to be shown in the Projection Space Views with the corresponding data subspace shown in the Data Space Views.

3.4.1 Projection Space Views:

Given a selected score function, these views look at the projected data based on the q-D random projections that maximize the selected score.

Score View:(Figure 2D) Our algorithm goes through a number of q-D random projections and ranks them based on their scores. The default views start off with the best scoring one. It is useful to toggle through the list of a few high-scoring random projections as each may have discovered a different view of the high-dimensional dataset showing the interesting structure the score captured. With visualization tools there is not one answer or one view but rather a number of views to look at different interesting slices or perspectives of the data. This barchart shows the scores of the top 10 scoring q-D random projections. This easily interpretable view functions as a controller to select a q-D random projection to investigate - selecting a bar updates all views.

Radar Plot Icon:(Figure 2E) The value of the chosen score function for a selected random projection subspace is listed under the Update button. However, the user might like to know the quality of each subspace with respect to the other score functions. Instead of listing the score values or providing another barchart, we use a simple visual icon to summarize the score values. The radar plot is a snapshot of the characteristics of the selected random projection

[1]Higher resolution images of all figures here are available at http://www.cs.uic.edu/ aanand/subspaceExplorer.html

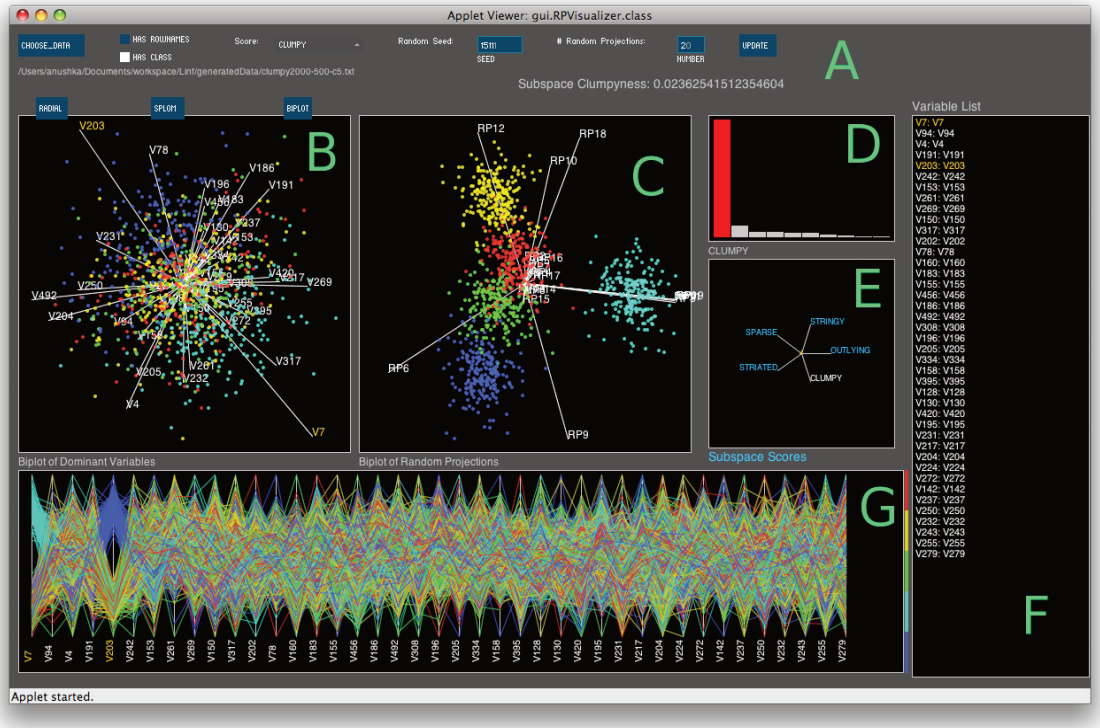

Figure 2: The Subspace Explorer showing a highly Clumpy 20-D random projection for a dataset with 2000 rows, 500 dimensions and clusters embedded in 5 dimensions. The detected known embedding dimensions are shown as orange text. (A) Control options. (B) Biplot View of the data subspace. (C) Random Projection View of the biplot of the projected data space. (D) ScoreView of the top 10 scoring random projections. The selected (red) bar represents one 20-D random projection which is shown in different perspectives in the other plots. (E) Radar plot Icon summary of all scores for the selected random projection. (F) Variable List (same set as B). (G) Parallel Coordinate View of the top used dimensions (same set as B)

on all the computed score functions. We chose to provide coarse information allowing for quick visual size-comparisons to not detract from the focus of analyzing the variables in the selected subspace.

Random Projection View:(Figure 2C) To see evidence of the interesting structure detected by the selected score function, we visualize the projected data matrix $\mathbf{B}_{n\times q} \in \mathbb{R}^q$. We could compute the MDS of the projected data but choose to compute the PCA and display a biplot. A biplot shows data observations as points and positions similar observations closer in the 2-D space. The vectors in the plot represent each 1-D random projection vector that make up the columns of the q-D space. The angle between the vectors indicates their similarity. Hovering over a random projection highlights the data variables it uses in the Data Space Views and clicking on a random projection updates the Data Space Views to show all the variables used in the selected 1-D random projection. This action allows the user to investigate one data subspace denoted by a subset of dimensions from the original data (for instance, Figure 12).

3.4.2 Data Space Views:

Given a selected q-D random projection and that we are using unit-weighted projections, a number of data dimensions with non-zero weights can be thought of as *turned on*. We count the number of times each dimension is turned on and select the minimum number of dimensions - of those used more than the median count or the thirty most commonly turned on - to display as the *top-used* ones. Showing more than thirty variables is not useful because of the clutter they introduce and, as often observed by statisticians, the intrinsic dimensionality of real datasets is usually fewer than ten dimensions, e.g., [30].

Radial View:(Figure 2B, Figure 3A) We are interested in the variables commonly used to capture a score and how they relate to each other as other work on dimensionality reduction [21]. To do so we use radial graph display of the top-used dimensions for the selected random projection. Variables are linked if they are used together in a 1-D random projection. The thickness of the line represents the number of co-occurences. The variables are ordered around the circle in a manner that minimizes the number of crossings. This ordering is done in a fast, greedy manner: a variable, v_1, is picked at random, the variable that co-occurs with it the most, v_2, is drawn next, the variable that co-occurs the most with v_2 is drawn next and so on. This way densely connected variables are positioned closer together. Hovering over a variable highlights its connections in cyan text labels and clicking on it updates the other Data Space Views to just show the data subspace of those linked variables. Figure 10 shows an instance where using this interaction allows the user to discover related variables that separate a group in the data.

Biplot View:(Figure 3C) To compare the data subspace with the projected data using the same visual metaphor, we construct a biplot of the data subspace. The top-used dimensions are selected from the original data matrix $\mathbf{X}$ to represent the data subspace. This view shows the similarity between data dimensions as they are drawn as vectors in the biplot with smaller angles indicating high similarity. This view often validates the structure described by the score function. Figure 2 is a interesting example where this is not the case. This data view shows observations collected in one big clump while the Random Projection View of the projected data shows multiple clumps which would obtain a higher clumpy score.

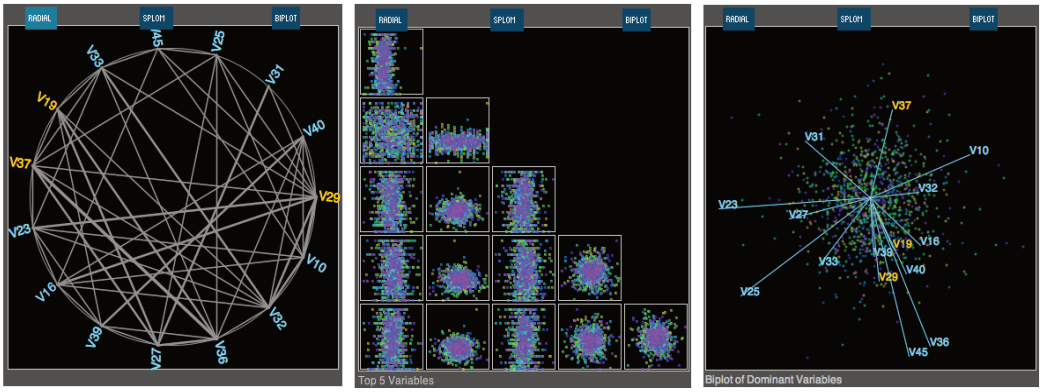

Figure 3: The interchangeable Data Views with the detected known embedding dimensions shown as orange text. (A) Radial View of links between all the top used dimensions in the selected 20-D random projection. (B) SPLOM View of the top 5 used dimensions. (C) Biplot View of all the top used dimensions.

Parallel Coordinate View:(Figure 2G) To learn finer detail about data distributions over variables we include a Parallel Coordinate Plot of the data observations across selected dimensions. The ordering of dimensions is consistent with the Radial View. While parallel coordinate plots are generally plagued by clutter, they are used to bring to light clusters within particular variables (see Figures 7 and 9)and to facilitate comparisons between data observations when used with interactivity in other linked plots. As we restrict the number of top-used variables to display, we curb the clutter from many dimensions.

SPLOM View:(Figure 3B) To aid the process of making connections between variables and displays, we provide a Scatterplot Matrix (SPLOM) showing the top 5 most commonly used variables. As with Jigsaw [38], different analysts wanted different views of the data. The SPLOM shows pairwise relationships between variables which highlights behavior that may not be evident in multidimensional projections so having this perspective provides a distinct view as shown in Figure 7. Selecting a scatterplot highlights the variables drawn on the x and y axes in all the other plots as the linking cascades.

Variable List:(Figure 2F) The long variable names in real datasets introduce clutter in the Data Space Views. To overcome this and still keep variable detail in the views, we use shortened identifiers in the views and provide a list of the mapping of original variable names to the variable identifiers here. We decided against using a tooltip providing localized detail in each plot to not overwhelm the user with many interaction associations in each plot.

4 Examples

The Subspace Explorer claims to find visual patterns using Scagnostic scores in subspaces of high-dimensional data. To validate the use of random projections in finding these interesting subspaces, we generate synthetic data with known structure and analyze subspaces discovered by our tool. Then we describe some findings when the Subspace Explorer is applied to real datasets.

4.1 Synthetic Data

We built a generator to create datasets with n observations and p ambient dimensions. It takes as input n, p and m, the number of dimensions representing the embedding space with the interesting structure (when appropriate). We describe the synthetic data generation process for each score function below. Figure 4 shows the result of our tests. For most scores, the figure shows the number of m embedding variables correctly discovered by the Subspace Explorer. As the number p increases and m is small, the ability to find most of the embedding space expectedly decreases.

OUTLYING

n	p	# Actual	#Actual Detected	#Total Detected
5000	500	79	53	79
5000	1000	79	41	79
5000	5000	75	24	75
2000	5000	63	19	63

CLUMPY

n	p	m	#Actual Detected	Visually Present
5000	500	5	3	Y
5000	1000	10	6	Y
5000	1000	50	19	Y
5000	5000	50	11	N

SPARSE

n	p	m	#Actual Detected	Visually Present
5000	500	20	14	Y
5000	1000	20	13	Y
5000	1000	50	26	Y
5000	5000	50	10	Y

STRINGY

n	p	m	#Actual Detected	Visually Present
5000	500	20	12	Y
5000	1000	20	6	Y
5000	1000	50	28	Y
5000	5000	50	8	Y

STRIATED

n	p	m	#Actual Detected	Visually Present
5000	500	20	11	Y
5000	1000	20	12	N
5000	1000	50	21	N
5000	5000	50	16	N

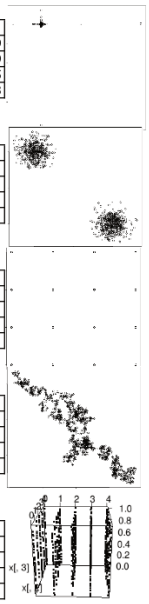

Figure 4: Table summarizing validation test results on synthetic data for each score function. The graphs on the right show canonical examples of the structures in 2-D that the score functions detect. Similar structures will be embedded in noisy higher-dimensional spaces.

Outlying To find outliers in mutlivariate datasets, we generate data in all p dimensions from the Cauchy distribution [25]. This distribution has extremely long tails. We seek to validate the outliers flagged by our multivariate outlier detection algorithm based on the MST in random projection space by comparing them to the ones detected by running through the original data matrix and finding those rows that differ from the sample mean by two sample standard deviations. The plots in Figure 5 show outliers flagged as slightly larger white points or white lines in the Parallel Coordinate View. These outliers were found in a dataset with $n = 5000, p = 500$. Figure 4 reports the number of actual outliers found in all the data, the number of outliers correctly detected by our algorithm along with the total number of outliers we detect for different datasets. The 2-D scatterplot shows a star-convex bivariate Cauchy distribution to give a picture of the outlying structure generated.

Clumpy We generate a clumpy dataset by first picking m dimen-

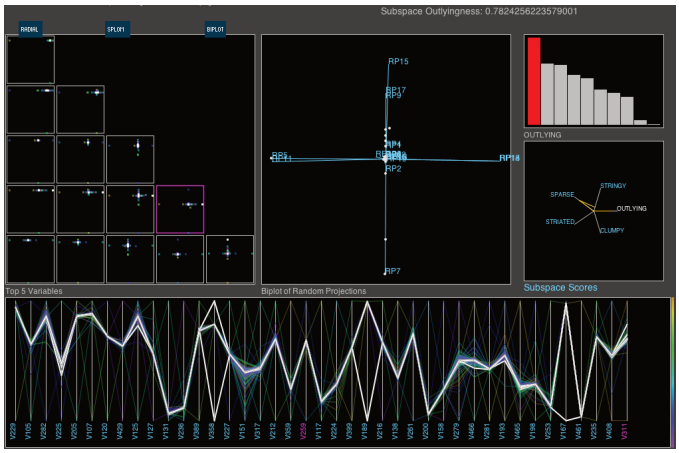

Figure 5: The Outlying score is maximized for multivariate Cauchy data and detected outliers are shown in white.

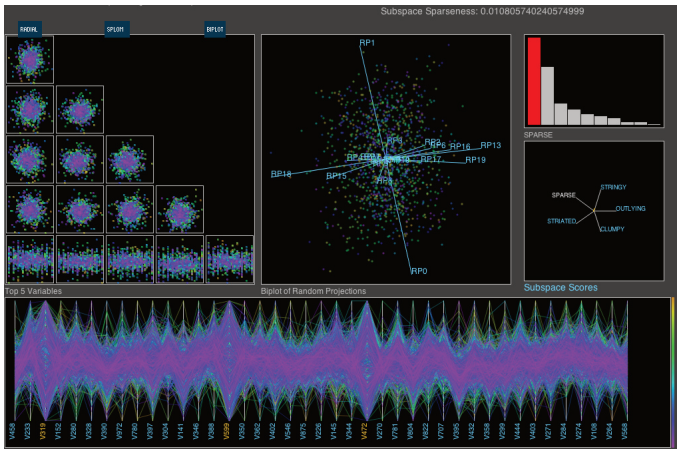

Figure 6: The Sparse score is maximized and correctly discovers three of the variables from the embedding space, shown in orange text.

sions out of the p at random. We place cluster centroids on the m vertices of a $(m-1)$-dimensional simplex. This means that a centroid at variable v_1 for instance, would have v_1 set to 1, the set of all the other $m-1$ variables set to 0 and the remaining $p-m$ variables to be set to small Gaussian random numbers. Given the centroids, we have to generate the remaining observations. To generate a data point, we randomly choose a cluster this point will belong to and we add to the centroid values small Gaussian noise. Figure 2 shows the Subspace Explorer with a dataset with $n = 2000, p = 500, m = 5$, so the clusters are embedded in 5 dimensions. The orange-colored variable names in the Data Space Views are two of the 5 variables from the known embedding space. It can be seen that this view clearly shows some clumpy sets of points which are colored by the cluster they belong to. It is important to note that simply performing PCA on all the data and projecting it onto the top two principle components does not reveal clusters or clumps. The 2-D scatterplot in Figure 4 shows the clusters generated at the vertices of a 1-D simplex when $m = 2$ and $p = 2$. In higher-dimensional embedding spaces (higher values of m), we would expect to see even more overlap of clusters.

Sparse To model sparseness, we first pick m dimensions out of the p at random. We set those m dimensions to have only ten unique values. The other $p-m$ dimensions are filled with small Gaussian noise. Flipping between Projection Space views to Data Space views can reveal subsets of the embedded space as shown in Figure 6. The Random Projection View shows some sparsity given the large number of noise variables in this dataset, which has $n = 5000, p = 1000, m = 50$. The number of embedding space variables discovered is shown in Figure 4. The scatterplot on the right shows how points in 2-D are distributed such that each variable puts points in four unique positions in the plane resulting in a 16-point lattice formation.

Striated For the Striated Scagnostic we were motivated by the artifact with the RANDU pseudo-random number generator that is known to generate points in 3-D that fall on a set of parallel 2-D planes [32]. For a RANDU dataset, there is a view in a narrow "squint angle" where the data are highly striated. We generated $m/2$ "lattice" variables with values selected from [0.0, 0.25, 0.5, 0.75, 1.0] with equal probabilities of .2. We then added $m/2$ "uniform" variables sampled from a uniform distribution. Finally, we added $p-m$ variables sampled from the same small Gaussians we used for the other simulations. Figure 7 shows a random projection with clear striation. However, as the number of Gaussian noise variables increase, the striped structure, which is seen when a tuple of lattice and uniform variables are selected together, is not as readily visually discernible as noted in Figure 4. The 3-D scatterplot on the right illustrates the kind of striped structure we create in the embedding space (here $m = p = 4$).

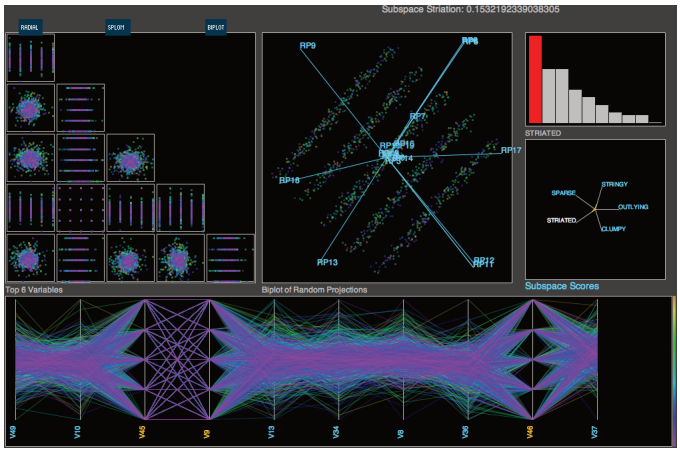

Figure 7: The Striated score is maximized showing a clearly striated view in this data set with $n = 5000, p = 50, m = 5$ where the embedding space is 10% of the ambient space..

Stringy We generate Stringy data by constructing smooth paths in m dimensions and embedding these in a p-dimensional space. Each observation will have a small Gaussian noise for the $p-m$ variables and for the m variables, we take their previous values and add a small amount of Gaussian noise, therefore introducing a first-order autocorrelation in the subspace. We see that the projected data biplot clearly shows a stringy shape for this data set with $n = 5000, p = 1000, m = 20$ in Figure 8 even though the embedding space accounts for only 2% of the data dimensionality. Again, the projection of all the data on the top two principle components completely obscures this pattern.

4.2 Real Data

We describe two scenarios using Subspace Explorer on publicly-available, large datasets to perform EDA.

Cancer:

Figure 9 shows a high-dimensional analysis not amenable to existing visual analytic platforms. There are 16,063 variables in this dataset [36], comprising genetic markers for 14 different types of

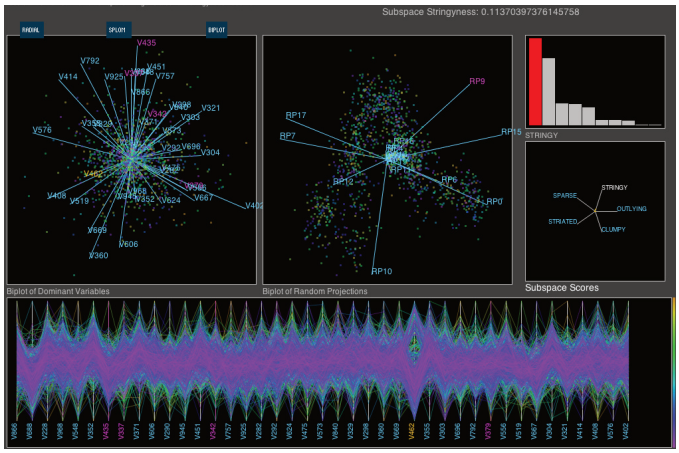

Figure 8: The Stringy score is maximized to show a clearly stringy view in the random projection space. Hovering over a random projection vector highlights the variables it turns on in magenta text.

Figure 9: The Cancer dataset has over 16,000 dimensions and 14 classes or types of cancer. We maximize the Clumpy score and select a scatterplot in the SPLOM showing good separation of the purple class.

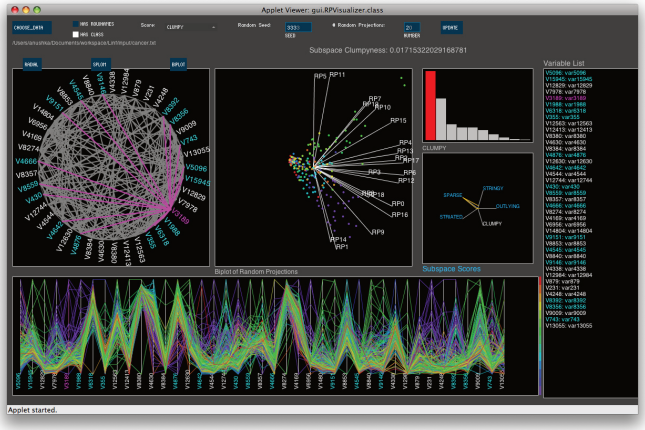

Figure 10: Hovering over a variable (highlighted in magenta) in the Radial View selects (in cyan) those it co-occurs with across the q-D random projections. Selecting it updates the Data Space Views with the data filtered on the variables co-occurring with the selected variable.

Figure 11: The GoMoos weather dataset has 160 dimensions from 10 different sensors. We maximize the Stringy score and select a scatterplot in the SPLOM showing two positively-correlated dimensions.

cancer. With wide datasets (large p), searching for the 2-D pair of variables that exhibit particular structure is a computationally expensive procedure on the order of $O(p^2)$. Subspace Explorer enables us to explore different perspectives of the data and to interactively learn about prominent markers related to certain classes of cancer. We highlight a scatterplot in the SPLOM View with variable *var*3189 on the y-axis, as this variable seems to separate class 13 (purple) from the other classes. Overall, the Random Projection View shows that views maximizing the Clumpy score push classes 13 and 5 (dark green) apart and separates them reasonably well from the other classes. We can hide the projection vectors from the Biplot by pressing the 'H' key and then hovering over data points to get details. Also, all these views are highly Sparse (as there are only 200 samples in this dataset).

Selecting variables that co-occur across the q-D random projections could direct users to variables that together describe a visual pattern. Here we are interested in finding dimensions that help differentiate particular cancer types. Figure 10 shows an instance where the Radial View is used to filter dimensions that are used together in the 1-D random projections that make up the q-D space. We find other variables that help separate class 13 like *var*3189. From this exploratory analysis one could conclude that classes 13 and 5 can be distinguished by subsets of gene markers and that peeling away those two classes might raise more informative views of the remaining classes to guide the selection of a classification model. Interestingly, this is the finding in the preliminary study of this data [36] where the authors conclude that pairwise distinctions can be made between select tumor classes using few genes, while multiclass distinctions among closely related tumor types are intrinsically more difficult. They build a multi-class Support Vector Machine classification strategy incorporating a one-vs-all scheme. Quick exploratory analysis through Subspace Explorer suggests a preliminary ordering of classes: classifying class 13 against everything, peeling those points away and then doing the same with class 5 points would enable the classifier to tackle the "easier" cases before the more mixed density classes.

Weather:

We collate hourly weather-sensor data from six buoys in the Gulf of Maine (http://gomoos.org/) from March 2002 to February 2008.

Figure 12: The GoMoos weather dataset with the Data Views showing the subspace described by the selected 1-D random projection RP8 (highlighted in magenta).

This results in a dataset with $n = 49,000$ and $p = 160$, as there are 16 variables measured at each of the 10 buoys. We are interested in exploring this data to discover dominant trends or relationships across the spatially-distributed sensors. Figure 11 shows a highly Stringy view of a random projection of the data. Generating a list of the top 10 Stringy views of this dataset takes less than 5 seconds. This Stringy appearance is due to the presence of a subset of smooth series within the data. The series that are relatively high-varying in this dataset are measures related to wind-direction, wind-gust, barometric pressure and visibility across the multiple buoy sensors. The barometric pressure readings have a large number of missing values across the buoys which accounts for the two clumps in the views of the data. The barometric pressure variables are also clearly discernible in the Parallel Coordinate View as having most of the data falling at the two extremes (for e.g. V97, V80). The SPLOM View allows the user to investigate correlated dimensions across buoys such as positively-correlated temperature readings or negatively-correlated density and temperature readings which constitute the smooth series in the data. Figure 12 shows the Data Views updated with just the data subspace from the selected 1-D random projection vector RP8. This Stringy subspace in the Biplot View is a projection of correlated variables related to wind speed and temperature across multiple buoys. The user can glean insight into natural groupings of variables by exploring the subspaces captured by different random projections and this can aid in model creation or feature selection.

4.3 Performance

Because we use efficient binning and random projections, the computation time for the scores based on the MST is $O(log_2(n)^2 q)$ where q defaults to 20. Therefore, we can run through a large number of random projections quickly to find views of the high-dimensional data that maximize our score functions and show interesting structure. On an Intel Core 2 Duo 2.2 GHz Apple Macbook Pro with 4 GB RAM running Java 1.6 and Processing 1.5.1, we run the Subspace Explorer on different large datasets and investigate its performance. The graphs in Figure 13 show computation time broken down into the time to read in the input data file, bin the n observations, generate a list of q-dimensional random projections from the p dimensional data, and score the list of projections by computing the MST and scores on the projected data. Empirical analysis shows that the time complexity of our approach is linear (with R^2 of 0.997 for the regression fit) in n and p with the dominant time taken to read in the data and bin it - both one pass operations. The

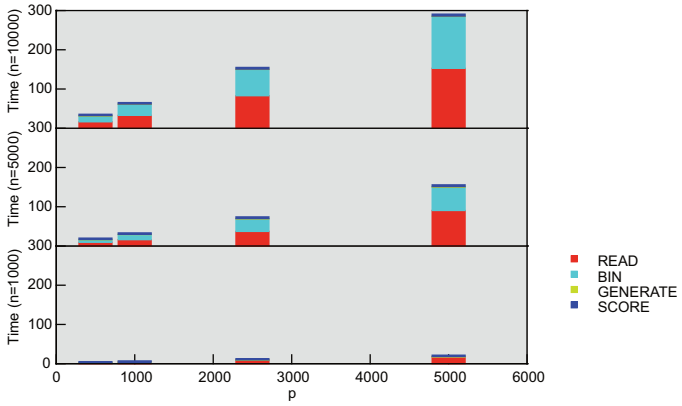

Figure 13: Computation times (in seconds) for large datasets where n is the number of observations and p is the number of dimensions.

largest file we read in this performance study is about 1 GB with $n = 10,000, p = 5000$. Thus, the Subspace Explorer is an $O(np)$ visual analytic explorer of subspaces.

5 Discussion & Conclusion

The Subspace Explorer was inspired by Projection Pursuit, but we must emphasize that it does not use the Friedman-Tukey algorithm. Our approach of iteratively scoring a large list of random projections to visualize high-scoring ones is efficient and ideal for visually exploring very high-dimensional datasets where Projection Pursuit or projections based on dense or sparse matrix decompositions is impractical. As an alternative, many have proposed searching for axis-parallel projections that exhibit "interesting" structure. While these orthogonal views of variables are often found and easy to interpret, as we mentioned in Section 2.1, axis-parallel views are not useful in finding high-dimensional structure. We look for good non-axis-parallel, affine, random projections that maximize a score, similar to a projection index, and visually compare the subspaces they represent. Also, we have described scores that go beyond the common ones that focus on finding views showcasing clusters and outliers. We have described a multivariate formulation of Scagnostic features to look for striated views where the data fall into stripes, sparse views where data fall into a few unique positions in space and stringy views where data exhibit some autocorrelation-type trend. We have identified five multivariate measures of visual structure and intend to pursue distilling additional ones. Furthermore, we are conducting user studies to investigate the intuitive use of our EDA dashboard and visual features to accomplish tasks related to feature selection.

The Subspace Explorer is a visual analytic platform facilitating the exploration of collections of random projections that maximize various score functions. The data analyst can drill-down to look at dimensions or variables that are indicators for a score function, therefore aiding in feature selection. This method of finding interesting views and identifying relevant dimensions in high-dimensional data is novel in that it is an implementation of iterated random projections and multivariate visual pattern scores wrapped up in a visual analytic platform.

Acknowledgements

This research is supported by NSF/DHS grant DMS-FODAVA-0808860.

References

[1] D. Achlioptas. Database-friendly random projections. In *Proc. of ACM SIGMOD*, pages 274–281. ACM, 2001.

[2] P. K. Agarwal, H. Edelsbrunner, and O. Schwarzkopf. Euclidean minimum spanning trees and bichromatic closest pairs. *Discrete and Computational Geometry*, pages 407–422, 1991.

[3] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proc. of ACM SIGMOD*, pages 94–105, 1998.

[4] G. Albuquerque, M. Eisemann, D. J. Lehmann, H. Theisel, and M. Magnor. Improving the visual analysis of high-dimensional datasets using quality measures. In *Proc. IEEE VAST 2010*, pages 19–26, 2010.

[5] G. Albuquerque, M. Eisemann, and M. Magnor. Perception-based visual quality measures. In *IEEE VAST*, pages 13–20, 2011.

[6] D. Asimov. The grand tour: A tool for viewing multidimensional data,. *Siam Journal on Scientific and Statistical Computing*, 6:128–143, 1985.

[7] C. Baumgartner, C. Plant, K. Kailing, H.-P. Kriegel, and P. Kroger. Subspace selection for clustering high-dimensional data. In *Proc. of the IEEE ICDM*, pages 11–18, 2004.

[8] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15:1373–1396, 2003.

[9] E. Bingham and H. Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Proc. of ACM SIGKDD*, pages 245–250, 2001.

[10] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proc. of COLT*, pages 144–152, New York, NY, USA, 1992. ACM.

[11] D. Caragea, D. Cook, V., and Honavar. Visual methods for examining support vector machine results, with applications to gene expression data analysis. Technical report, Iowa State University, 2005.

[12] J. Choo, S. Bohn, and H. Park. Two-stage framework for visualization of clustered high dimensional data. In *IEEE VAST*, pages 67–74, 2009.

[13] D. Cook, A. Buja, J. Cabrera, and C. Hurley. Grand tour and projection pursuit. *Journal of Computational and Graphical Statistics*, 4:155–172, 1995.

[14] X. Z. Fern and C. E. Brodley. Random projection for high dimensional data clustering: A cluster ensemble approach. In *ICML'03*, pages 186–193, 2003.

[15] D. Fradkin and D. Madigan. Experiments with random projections for machine learning. In *Proc. of ACM KDD*, pages 517–522, New York, NY, USA, 2003. ACM.

[16] J. H. Friedman and J. W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, C-23:881–890, 1974.

[17] J. A. Hartigan and S. Mohanty. The runt test for multimodality. *Journal of Classification*, 9:63–70, 1992.

[18] C. Hegde, M. Wakin, and R. Baraniuk. Random projections for manifold learning. In *Neural Information Processing Systems (NIPS)*, Vancouver, BC, 2007.

[19] A. Hinneburg, D. Keim, and M. Wawryniuk. Using projections to visually cluster high-dimensional data. *Computing in Science Engineering*, 5(2):14 – 25, 2003.

[20] H. Hofmann, K. Kafadar, and H. Wickham. Letter-value plots: Boxplots for large data. *The American Statistican*, Submitted.

[21] S. Ingram, T. Munzner, V. Irvine, M. Tory, S. Bergner, and T. Mller. Dimstiller: Workflows for dimensional analysis and reduction. In *IEEE VAST'10*, pages 3–10, 2010.

[22] S. J. and B. Shneiderman. A rank-by-feature framework for interactive exploration of multidimensional data. *Information Visualization*, 4:96–113, March 2005.

[23] D. H. Jeong, C. Ziemkiewicz, B. Fisher, W. Ribarsky, and R. Chang. iPCA: An Interactive System for PCA-based Visual Analytics. *Computer Graphics Forum*, 28(3):767–774, 2009.

[24] L. O. Jimenez and D. A. Landgrebe. Projection pursuit for high dimensional feature reduction: paralleland sequential approaches. In *Geoscience and Remote Sensing Symposium*, volume 1, pages 148–150, 1995.

[25] N. L. Johnson, S. Kotz, and N. Balakrishnan. Chapter 16. In *Continuous Univariate Distributions*, volume 1. Wiley & Sons, 1994.

[26] W. B. Johnson and J. Lindenstrauss. Lipschitz mapping into Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

[27] D. A. Keim. Designing pixel-oriented visualization techniques: Theory and applications. *IEEE TVCG*, 6(1), 2000.

[28] J. Kruskal and R. Shepard. A nonmetric variety of linear factor analysis. *Psychometrika*, 39:123–157, 1974.

[29] E.-K. Lee, D. Cook, S. Klinke, and T. Lumley. Projection pursuit for exploratory supervised classification. *Journal of Computational and Graphical Statistics*, 14:831–846, 2005.

[30] E. Levina and P. J. Bickel. Maximum likelihood estimation of intrinsic dimension. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 777–784, Cambridge, MA, 2005. MIT Press.

[31] P. Li, T. J. Hastie, and K. W. Church. Very sparse random projections. In *Proc. of ACM KDD*, pages 287–296, 2006.

[32] G. Marsaglia. Random numbers fall mainly in the planes. In *Proc. National Academy of Sciences*, volume 61, pages 25–28, 1968.

[33] V. Osinska and P. Baia. Nonlinear approach in classification visualization and evaluation. In *Congress ISKO, '09*, pages 257–270, 2009.

[34] L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: a review. *SIGKDD Explor. Newsl.*, 6(1):90–105, June 2004.

[35] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572, 1901.

[36] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J. Mesirov, T. Poggio, W. Gerald, M. Loda, E. Lander, and T. Golub. Multiclass cancer diagnosis using tumor gene expression signature. *PNAS*, 98:15149–15154, 2001.

[37] M. Sips, B. Neubert, J. P. Lewis, and P. Hanrahan. Selecting good views of high-dimensional data using class consistency. *Computer Graphics Forum*, 28(3):831–838, 2009.

[38] J. Stasko, C. Gorg, Z. Liu, and K. Singhal. Jigsaw: Supporting investigative analysis through interactive visualization. In *Proc. of the IEEE VAST*, pages 131–138, 2007.

[39] W. Stuetzle. Estimating the cluster tree of a density by analyzing the minimal spanning tree of a sample. *Journal of Classification*, 20:25–47, 2003.

[40] H. A. Sturges. The choice of a class interval. *Journal of the American Statistical Association*, 21:65–66, 1926.

[41] A. Tatu, G. Albuquerque, M. Eisemann, P. Bak, H. Theisel, M. Magnor, and D. Keim. Automated analytical methods to support visual exploration of high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics*, 17:584–597, 2011.

[42] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[43] A. N. Tikhonov and V. Y. Arsenin. *Solutions of ill posed problems.* V. H. Winston and Sons, Wiley, NY, 1977.

[44] J. W. Tukey. Mathematics and the picturing of data. In *Pro. of the International Congress of Mathematicians*, pages 523–531, Vancouver, Canada, 1974. Canadian Mathematical Congress.

[45] C. Turkay, P. Filzmoser, and H. Hauser. Brushing dimensions – a dual visual analysis model for high-dimensional data. *IEEE TVCG*, 17(12):2591–2599, 2011.

[46] L. Wilkinson, A. Anand, and T. Dang. Chirp: A new classifier based on composite hypercubes on iterated random projections. In *ACM KDD*, 2011.

[47] L. Wilkinson, A. Anand, and R. Grossman. Graph-theoretic scagnostics. In *Proc. of the IEEE Information Visualization 2005*, pages 157–164. IEEE Computer Society Press, 2005.

[48] 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.

[49] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15:2006, 2004.