Grand Tours, Projection Pursuit Guided Tours and Manual Controls

Dianne Cook[1], Andreas Buja[2], Eun-Kyung Lee[3], and Hadley Wickham[4]

[1] Department of Statistics, Iowa State University, `dicook@iastate.edu`
[2] Statistic Department, The Wharton School, University of Pennsylvania, `buja@wharton.upenn.edu`
[3] Bioinformatics and Biostatistics Laboratory, Department of Statistics, Seoul National University, `gracesle@snu.ac.kr`
[4] Department of Statistics, Iowa State University, `hadley@iastate.edu`

1 Introductory Notes

How do we find structure in multidimensional data when computer screens are only two-dimensional? One approach is to project the data onto one or two dimensions.

Projections are used in classical statistical methods like principal component and linear discriminant analysis. Principal component analysis (eg Johnson & Wichern (2002)) chooses a projection to maximize the variance. Fisher's linear discriminant (eg Johnson & Wichern (2002)) chooses a projection that maximizes the relative separation between group means. Projection pursuit (eg Huber (1985)) generalizes these ideas into a common method, where an arbitrary function on projections is optimized. The scatterplot matrix (eg Becker & Cleveland (1987)) can also be considered to be a projection method. It shows projections of the data on all pairs of coordinate axes, the two-dimensional marginal projections of the data. These projection methods choose a few select projections out of infinitely many projections.

What is hidden from the user who views only a few static projections? There could be a lot. The reader may be familiar with an ancient fable from India about the blind men and the elephant. One grabbed his tail and swore the creature was a rope. Another felt the elephant's ear and yelled it was a hand fan. Yet another grabbed his trunk and exclaimed he'd found a snake. They argued and argued about what the elephant was like, until a wise man settled the fight. They were all right, each describing different parts of the elephant. Looking at a few static projections of multivariate data is like the blind men feeling parts of the elephant and inferring the nature of the whole beast.

How can a more systematic presentation of all possible projections be constructed? Static projections can be strung together into a movie using

interpolation methods, providing the viewer with an overview of multivariate data. These interpolation methods are commonly called tours. They provide a general approach to choose and view data projections, allowing the viewer to mentally connect disparate views, and thus supporting the exploration of a high-dimensional space. We use tours to explore multivariate data like we might explore a new neighborhood: we could walk randomly to discover unexpected sights, employ a guide, or guide ourselves using a map. These modes of exploration correspond to three commonly available types of tours, available in the software, GGobi (Swayne, Temple Lang, Cook & Buja 2001-), which is used in this chapter to illustrate the methods.

- In the *grand tour*, we walk randomly around the landscape discovering unexpected sights – the grand tour shows all projections of the multivariate data. This requires time and we may spend a lot of time wandering around boring places and miss the highlights.
- Using a *projection pursuit guided tour*, we employ a tour guide who takes us to the features that they think are interesting. We improve the probability of stopping by the interesting sights by selecting more views that are "interesting" based on a projection pursuit index.
- *Manual control* takes the steering wheel back from the guide, enabling the tourist to decide on the next direction. We choose a direction by controlling the projection coefficient for a single variable. This allows us to explore the neighborhood of an interesting feature or to understand the importance of a variable on the feature.

1.1 Some Basics on Projections

What is a projection? We can think of a projection as the shadow of an object. Especially if it is a two-dimensional projection, then the projection is the shadow the object casts under a bright light (Figure 1). If the object rotates in the light, we see many different two-dimensional shadows and we can infer the shape of the object itself.

Mathematically, projected data is computed by multiplying an $n \times p$ dimensional data matrix, $\mathbf{X}$, having n sample points in p dimensions, by an orthonormal $p \times d$ dimensional projection matrix, $\mathbf{A}$. The dimension of the space we are projecting onto is d. For example, to project a three-dimensional object (3 columns, or variables, of data) onto a two-dimensional plane (the shadow of the object) we would use an orthonormal 3×2 matrix.

Using numbers, suppose our data matrix and projection were these:

Fig. 1. Projections are like shadows. They are useful for obtaining a sense of the shape of a data set, when many projections are viewed. What may look like a horse in one projection may be revealed as carefully oriented pair of hands by another projection.

$$\mathbf{X} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 15 \\ 0 & 15 & 0 \\ 0 & 15 & 15 \\ 15 & 0 & 0 \\ 15 & 0 & 15 \\ 15 & 15 & 0 \\ 15 & 15 & 15 \end{bmatrix}_{8\times3} \quad \text{and} \quad \mathbf{A}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}_{3\times2} \quad \text{then} \quad \mathbf{XA}_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 15 \\ 0 & 15 \\ 15 & 0 \\ 15 & 0 \\ 15 & 15 \\ 15 & 15 \end{bmatrix}_{8\times2}$$

is the first two columns of the data matrix. If instead

$$\mathbf{A}_2 = \begin{bmatrix} 0.71 & -0.42 \\ 0.71 & 0.42 \\ 0 & 0.84 \end{bmatrix}_{3\times2} \quad \text{then} \quad \mathbf{XA}_2 = \begin{bmatrix} 0 & 0 \\ 0 & 12.60 \\ 10.65 & 6.30 \\ 10.65 & 18.90 \\ 10.65 & -6.30 \\ 10.65 & 6.30 \\ 21.30 & 0 \\ 21.30 & 12.60 \end{bmatrix}_{8\times2}$$

a combination of all three variables.

These projections are illustrated in Figure 2. The top row shows the data projections, $\mathbf{XA}_1, \mathbf{XA}_2$ respectively. The bottom row displays the projection coefficients, $\mathbf{A}_1, \mathbf{A}_2$. Each row of the projection matrix, represented by a line in this display, determines the contribution each variable makes to the projected data view. Each row can be considered to be the projection of the p-dimensional coordinate axis for a variable. In $\mathbf{A}_1$ the data projection is constructed from purely variable 1 in the horizontal direction, and variable 2 in the vertical direction. In $\mathbf{A}_2$ variables 1 and 2 share the horizontal direction, and variable 3 makes no contribution horizontally. Vertically all three variables make a contribution, but variable 3 has twice the contribution of the other two variables. This type of axis display is used to match structure in a data projection with the variables in the data, and hence enable to the analyst to interpret the data.
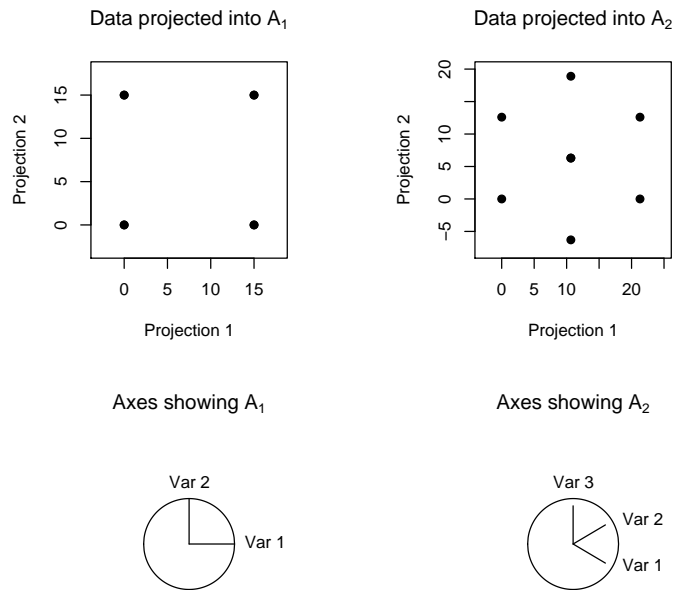
Fig. 2. Two two-dimensional data projections. The two top plots are the data projections, and the two bottom plots are illustrations of the projection coefficients.

We also commonly use one-dimensional projections in data analysis. The one-dimensional projected data is typically displayed as a dotplot, histogram or density plot. Consider the two-dimensional data in Figure 3 (left plot), and two one-dimensional projections (middle, right). The projection matrices are:

$$\mathbf{A}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{and} \quad \mathbf{A}_2 = \begin{bmatrix} -\frac{\sqrt{3}}{2} \\ -\frac{1}{2} \end{bmatrix}$$
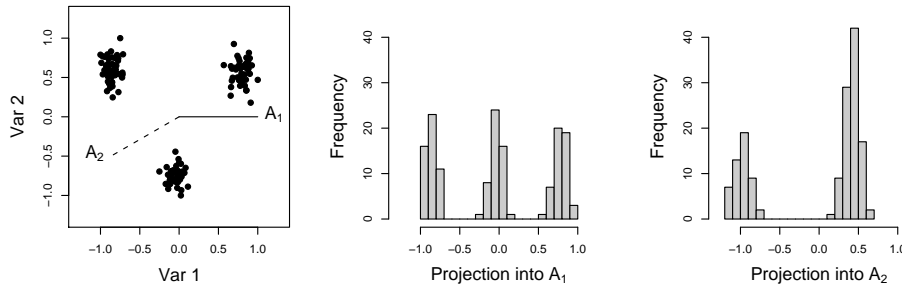
respectively.

Fig. 3. Two one-dimensional projections of two-dimensional data.

1.2 What Structure is Interesting?

When we use tours, what are we looking for in data? Anything that is not normally distributed, a little unusual or unexpected. For example, clusters of points, outliers, linear dependencies, non-linear relationships, and low-dimensional sub-structures. All of these can be present in multivariate data but hidden from the viewer who only chooses a few static projections. Figures 4 and 5 show some examples.

In Figure 4 a scatterplot matrix of all pairwise plots is shown at left, and a tour projection is shown at right. The pairwise plots show some linear association between three variables, particularly between the variables TEMP and PRESS, and TEMP and CODE. However, viewing the data in a tour reveals that the three variables are really perfectly related, with perhaps a slight nonlinear association. The projection of the data revealing the perfect relationship is:

$$\mathbf{A} = \begin{bmatrix} \text{TEMP} & -0.720 & 0.470 \\ \text{PRESS} & -0.668 & -0.671 \\ \text{CODE} & -0.191 & 0.573 \end{bmatrix}$$

In Figure 5 the pairwise scatterplots (at left) suggest there is some clustering of the data points in this six variable data set. The tour projection (right) reveals three well-separated clusters. The projection revealing the clusters is:
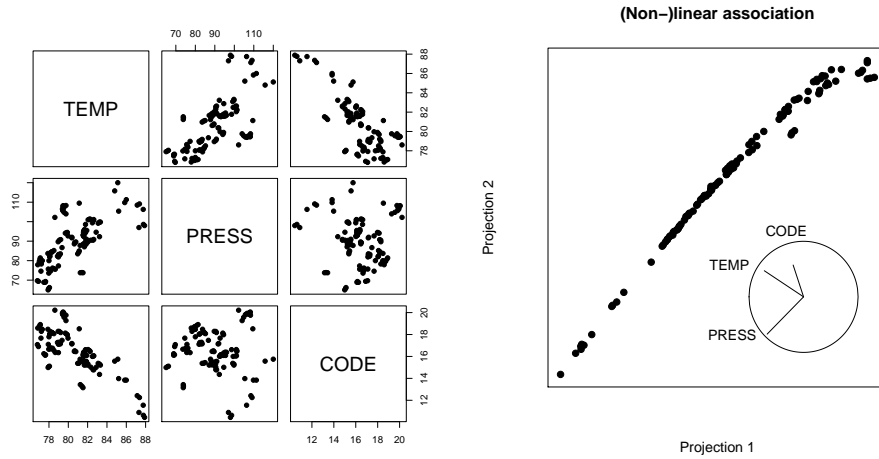
Fig. 4. The three variables show some association in the scatterplot matrix (all pairwise marginal projections in left plot), but they are revealed to be perfectly related by a tour (right plot).
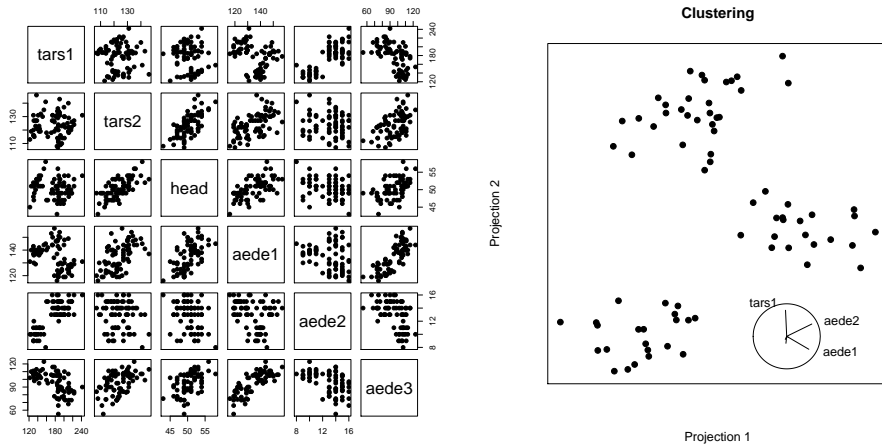

Fig. 5. The six variables show some clustering in the scatterplot matrix (all pairwise marginal projections in left plot), but the three clusters are better separated in a tour projection (right plot).

$$\mathbf{A} = \begin{bmatrix} \text{tars1} & -0.035 & 0.801 \\ \text{tars2} & -0.023 & -0.215 \\ \text{head} & 0.053 & -0.032 \\ \text{aede1} & 0.659 & -0.398 \\ \text{aede2} & 0.748 & 0.378 \\ \text{aede3} & -0.043 & -0.097 \end{bmatrix}$$

which is primarily a combination of three of the six variables, tars1, aede1, aede3.

2 Tours

Most of us are familiar with three-dimensional rotation. This is something we can do in the real world, take an object and rotate it with our hands we typically, or walk around an object to view it from all sides. Views of p-dimensional data can be computed in analogous ways, by rotating the entire p-dimensional data (Wegman 1991, Carr, Wegman & Luo 1996, Tierney 1991), or taking $d < p$-dimensional projections of p-dimensional data which is like looking at it from different sides. Viewing the data from different sides corresponds to rotating the d-dimensional projection plane rather than the data. Rotating the projection plane is computed by interpolating between a starting plane and a target plane. This is the method used in GGobi. It is more complicated to compute but it has some desirable properties, primarily that within-plane spin is eliminated. Planar rotations are discussed in detail in Asimov (1985), more simply in Buja & Asimov (1986), and very technically in Buja, Cook, Asimov & Hurley (2005), and also Asimov & Buja (1994), Buja, Hurley & McDonald (1986).

2.1 Terminology: Plane, Basis, Frame, Projection

A p-dimensional Euclidean space is described by a p-dimensional orthonormal basis, which conventionally is:

$$\begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & & \\ \vdots & & \ddots & \\ 0 & & & 1 \end{bmatrix}_{p \times p}$$

but any p-dimensional orthonormal basis could be used. There are infinitely many possibilities.

A d-dimensional plane contained in the p-space is defined by a d-dimensional orthonormal frames (or bases or projection). For example, consider the $d = 2$-dimensional frames:

$$\mathbf{A}_1 = \begin{bmatrix} \mathbf{a}_{11} \\ \mathbf{a}_{12} \\ \vdots \\ \mathbf{a}_{1p} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{A}_2 = \begin{bmatrix} \mathbf{a}_{21} \\ \mathbf{a}_{22} \\ \vdots \\ \mathbf{a}_{2p} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix}$$

which both describe the same two-dimensional plane in p-space. We conventionally use $\mathbf{A}_1$ to describe the two-dimensional plane, but we could just as validly use $\mathbf{A}_2$. (There are infinitely many bases that describe the same plane.) Figure 6 illustrates the two bases, which result in the same but rotated views of the data.
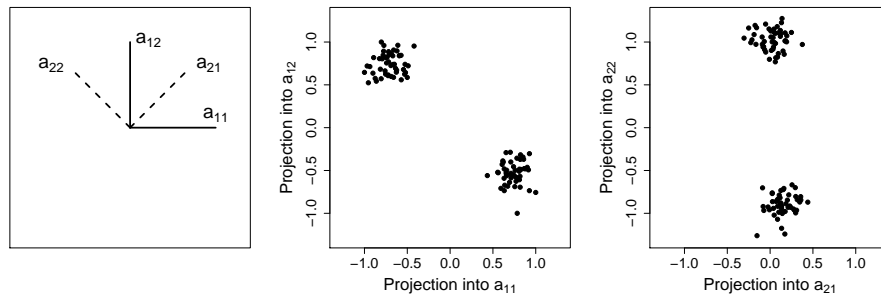

Fig. 6. Two different bases that describe the same plane, and the resulting rotated views of the data.

2.2 Interpolating Between Projections: Making a Movie

Creating a movie of data projections has two parts: choosing a new target plane and interpolating along a geodesic path from the current (starting) plane to the new plane. (Geodesic means sphere. The interpolation conserves the orthonormality of the projection vectors from step to step.) The frame that describes the starting plane is carried through the sequence of intermediate planes, preventing the data from rotating within the plane of view. That is, we avoid doing a rotation of the data like that in Figure 6. This type of within-plane rotation is distracting to the viewer, akin to viewing a scene while standing on a wobbly platform. All rotation occurs out of the viewing plane. The algorithm that is used in GGobi creates an interpolation between two planes and follows these steps:

1. Given a starting $p \times d$ projection $\mathbf{A}_a$, describing the starting plane, create a new target projection $\mathbf{A}_z$, describing the target plane. The projection

may also be called an orthonormal frame. To find the optimal rotation of the starting plane into the target plane we need to find the frames in each plane which are the closest.

2. Determine the shortest path between frames using singular value decomposition. $\mathbf{A}'_a\mathbf{A}_z = \mathbf{V}_a \Lambda \mathbf{V}'_z, \quad \Lambda = \mathrm{diag}(\lambda_1 \geq \ldots \geq \lambda_d)$, and the principal directions in each plane are $\mathbf{B}_a = \mathbf{A}_a\mathbf{V}_a, \mathbf{B}_z = \mathbf{A}_z\mathbf{V}_z$. The principal directions are the frames describing the starting and target planes which have the shortest distance between them. The rotation is defined with respect to these principal directions. The singular values, $\lambda_i, i = 1, \ldots, d$, define the smallest angles between the principal directions.

3. Orthonormalize $\mathbf{B}_z$ on $\mathbf{B}_a$, giving $\mathbf{B}_*$, to create a rotation framework.

4. Calculate the principal angles, $\tau_i = \cos^{-1}\lambda_i, i = 1, \ldots, d$.

5. Rotate the frames by dividing the angles into increments, $\tau_i(t)$, for $t \in (0, 1]$, and create the ith column of the new frame, $\mathbf{b}_i$, from the ith columns of $\mathbf{B}_a$ and $\mathbf{B}_*$, by $\mathbf{b}_i(t) = \cos(\tau_i(t))\mathbf{b}_{ai} + \sin(\tau_i(t))\mathbf{b}_{*i}$. When $t = 1$, the frame will be $\mathbf{B}_z$.

6. Project the data into $\mathbf{A}(t) = \mathbf{B}(t)\mathbf{V}_a$.

7. Continue the rotation until $t = 1$. Set the current projection to be $\mathbf{A}_a$ and go back to step 1.

2.3 Choosing the Target Plane

Grand Tour

The default method for choosing the new target plane is to use a random sequence. It is called a grand tour. The sequence is determined by randomly selecting a frame describing a target plane from the space of all possible projections. A grand tour may be considered to be an interpolated random walk over the space of all planes.

A target frame is chosen randomly by standardizing a random vector from a standard multivariate normal distribution. Sample p values from a standard univariate normal distribution, resulting in a sample from a standard multivariate normal. Standardizing this vector to have length equal to one gives a random value from a $(p-1)$-dimensional sphere, that is, a randomly generated projection vector. Do this twice to get a two-dimensional projection, where the second vector is orthonormalized on the first.

Figure 7 illustrates the tour path, using GGobi to look at itself. Using GGobi, we recorded the sequence of 9000 1D projections displayed of three-dimensional data. This tour path is a set of 9000 points on a three-dimensional sphere, where each point corresponds to a projection. We use a tour to view the path (top row of plots). The starting projection is $\mathbf{A}' = (1\ 0\ 0)$, indicated by a large point in the display. It is on the center right in the left plot, a projection in the first two variables. The corresponding data projection is shown at right. The grand tour path zig zags around the three-dimensional sphere. With enough time it will entirely fill the surface of the three-dimensional sphere.
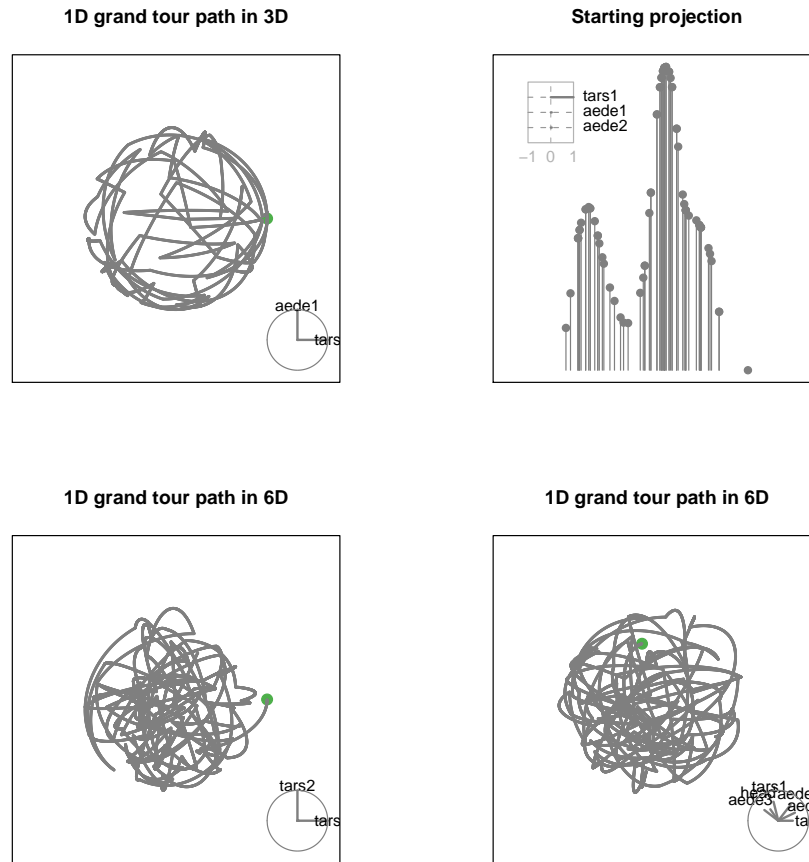
Fig. 7. Some views of one-dimensional grand tour paths in three-dimensional (top) and 6-dimensional (bottom). The path consists of a sequence of points on a three-dimensional and 6-dimensional sphere respectively. Each point corresponds to a projection from three-dimensional or 6-dimensional to one-dimensional.

The bottom row of plots shows two views of a grand tour path of 20000 one-dimensional projections of six-dimensional data.

Projection Pursuit Guided Tour

In a projection pursuit guided tour (Cook, Buja, Cabrera & Hurley 1995) the next target basis is selected by optimizing a function defining interesting projections. The same interpolation algorithm used in the grand tour is used to smoothly change between the current projection and the target.

Used alone, projection pursuit (PP) seeks out low dimensional projections that expose interesting features of the high dimensional point cloud. It does this by optimizing a criterion function, called the projection pursuit index, over all possible d-dimensional projections of p-dimensional data,

$$\max f(\mathbf{XA}) \quad \forall \mathbf{A}$$

subject to orthonormality constraints on $\mathbf{A}$. Projection pursuit results in a number of static plots of projections which are deemed interesting. This contrasts the dynamic movie of projections provided by a grand tour. Combining the two in an interactive framework (guided tour) provides the interesting views in the context of surrounding views, allowing better structure detection and better interpretation of structure.

The optimization procedure is an important part of a projection pursuit guided tour. The purpose of PP optimization is to find all of the interesting projections, not only to find one global maximum, because often the local maxima can reveal unexpectedly interesting data structures. For this reason, the PP optimization algorithm needs to be flexible enough to find global and local maxima.

Posse (1990) compared several optimization procedures, and suggest a random search for finding the global maximum of a projection pursuit index. Cook et al. (1995) use a derivative-based hill-climbing optimization, which when merged with a grand tour is a lot like an interactive simulated annealing algorithm. It finds local maximum as well as global maxima. Klein and Dubes (1989) showed that simulated annealing can produce good results for PP.

Lee, Cook, Klinke & Lumley (2005) use the modified simulated annealing method. They use two different temperatures, one for neighborhood definition, and the other guards against getting trapped into a local minimum allowing the algorithm to visit a local minimum and then jump out and explore for other minima. The temperature for neighborhood definition is re-scaled by a predetermined cooling parameter defined before starting. This cooling parameter determines how many iterations are needed to converge and whether the maximum is likely to be a local maximum or a global maximum. The initial temperature for the probability also controls the speed of convergence.

The optimization algorithm follows these steps:

1. From the current projection, $\mathbf{A}_a$, calculate the initial projection pursuit index value, I_0.
2. For $i = 1, 2, \cdots$
    a) Generate a new projections, $\mathbf{A}_i = \mathbf{A}_a + c^i \mathbf{B}$, from a neighborhood of the current projection where the size of the neighborhood is specified by the cooling parameter, c, and a random projection, $\mathbf{B}$.
    b) Calculate the index value for the new projection, I_i, the difference, $\triangle I_i = I_i - I_0$ and $T_i = \frac{T_0}{\log(i+1)}$, where T_0 is an initial temperature

    c) Take the new projection $\mathbf{A}_i$ to be the target, $\mathbf{A}_z$ with probability $\rho = \min(\exp\left(\frac{\triangle I_i}{T_i}\right), 1)$ and interpolate from the current projection $\mathbf{A}_a$ to the target projection $\mathbf{A}_z$

    d) Set $\mathbf{A}_a = \mathbf{A}_i$

3. Repeat (a) - (d) until $\triangle I_i$ is small.

Figure 8 (top two plots) shows a PP guided tour path (one-dimensional projections from three-dimensional space). It looks very similar to a grand tour path, but there is a big difference: the path repeatedly returns to the same projection, and its opposite (highlighted). The middle plot traces the PP index value over time. The path iterates between optimizing the PP function, and random target basis selection. The peaks (highlighted) are the maxima of the PP index, and for the most part, these are at the same projection. The corresponding data projections (the positive and negative of the same vector) are shown in the bottom row. The index is responding to a bimodal pattern in the data.

There are numerous PP indices. Here are a few that are used in GGobi. For simplicity in these formula, it is assumed that $\mathbf{X}$ is sphered, has mean zero and variance-covariance equal to the identity matrix.

Holes:

$$I_{Holes}(\mathbf{A}) = \frac{1 - \frac{1}{n}\sum_{i=1}^{n}\exp(-\frac{1}{2}\mathbf{y}_i'\mathbf{y}_i)}{1 - \exp(-\frac{d}{2})}$$

where $\mathbf{X}\mathbf{A} = \mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n]^T$ is a $n \times d$ matrix of the projected data.

Central Mass:

$$I_{CM}(\mathbf{A}) = \frac{\frac{1}{n}\sum_{i=1}^{n}\exp(-\frac{1}{2}\mathbf{y}_i'\mathbf{y}_i) - \exp(-\frac{d}{2})}{1 - \exp(-\frac{d}{2})}$$

where $\mathbf{X}\mathbf{A} = \mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n]^T$ is a $n \times d$ matrix of the projected data.

LDA:

$$I_{LDA}(\mathbf{A}) = 1 - \frac{|\mathbf{A}'\mathbf{W}\mathbf{A}|}{|\mathbf{A}'(\mathbf{W} + \mathbf{B})\mathbf{A}|}$$

where $\mathbf{B} = \sum_{i=1}^{g} n_i(\bar{\mathbf{X}}_{i.} - \bar{\mathbf{X}}_{..})(\bar{\mathbf{X}}_{i.} - \bar{\mathbf{X}}_{..})'$, $\mathbf{W} = \sum_{i=1}^{g}\sum_{j=1}^{n_i}(\mathbf{X}_{ij} - \bar{\mathbf{X}}_{i.})(\mathbf{X}_{ij} - \bar{\mathbf{X}}_{i.})'$ are the "between" and "within" sum of squares matrices from linear discriminant analysis, g is the number of groups, $n_i, i = 1, ..., g$ is the number of cases in each group.
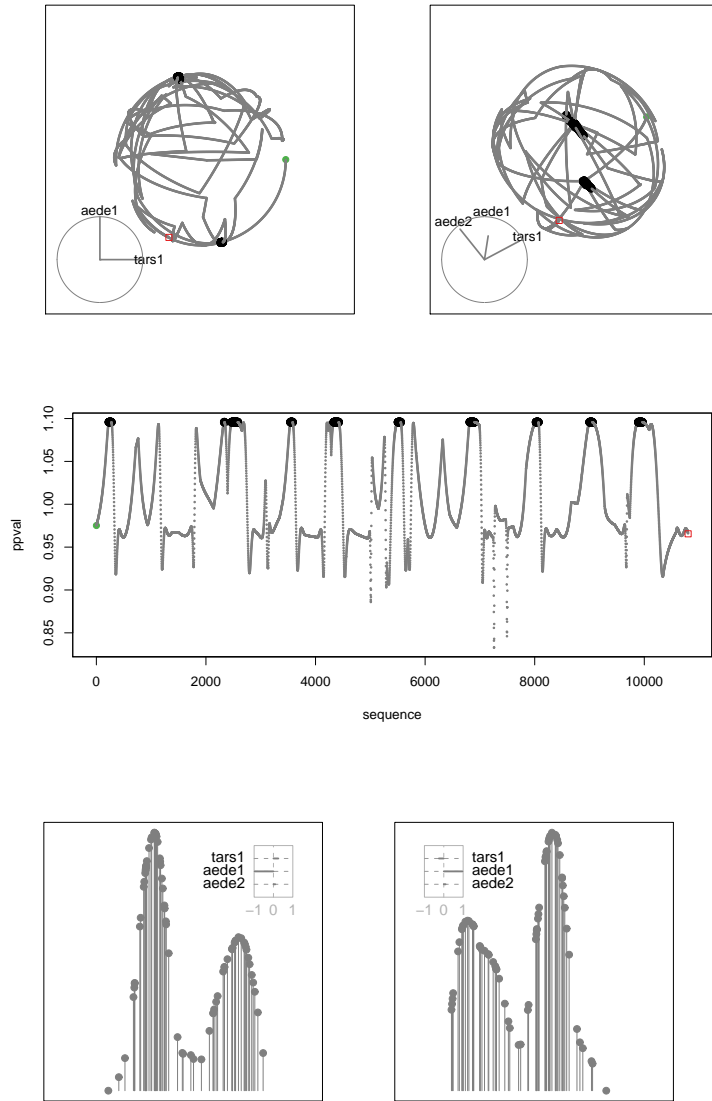
Fig. 8. Projection pursuit guided tour path.

PCA: This is only defined for $d = 1$.

$$I_{PCA} = \frac{1}{n}\mathbf{Y}'\mathbf{Y} = \frac{1}{n}\sum_{i=1}^{n} y_i^2$$

where $\mathbf{XA} = \mathbf{Y} = [y_1, y_2, \cdots, y_n]^T$ is a $n \times d$ matrix of the projected data.
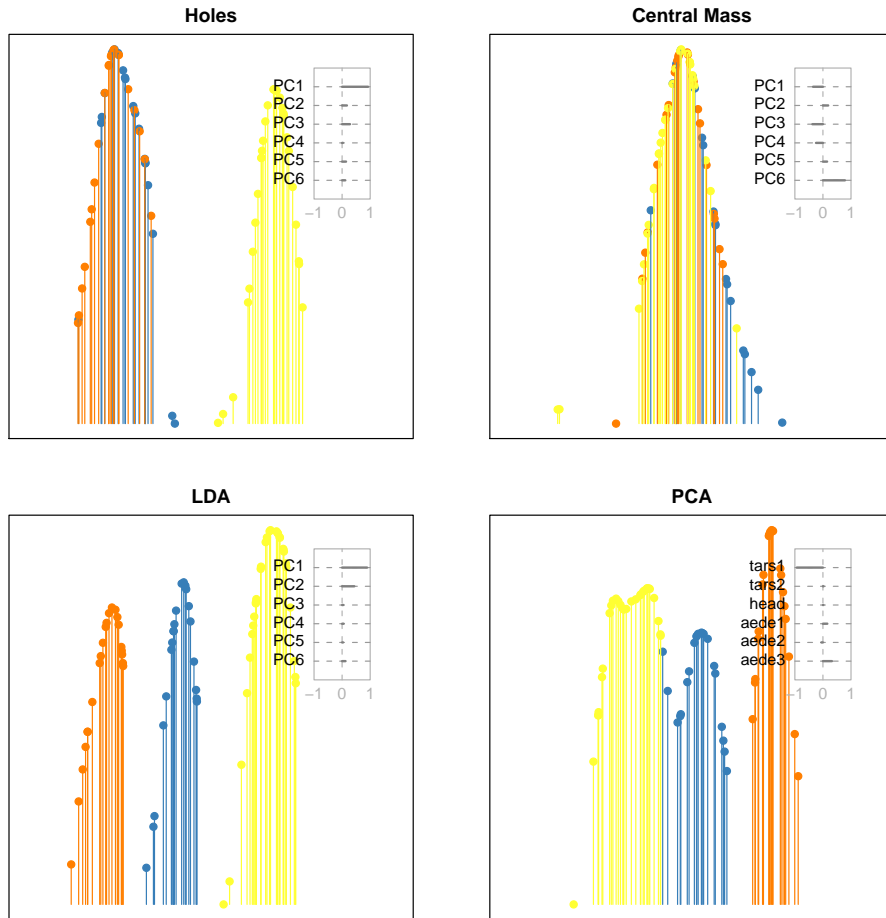

Fig. 9. One-dimensional data projections corresponding to maxima, from four different PP indices computed on the same data.

Figure 9 shows the results of using different indices on the same data. The Holes index finds a projection where there is a gap between two clusters of points. The Central Mass index finds a projection where a few minor

outliers are revealed. The LDA index finds a projection where three clusters can be seen. The PCA index finds a tri-modal data projection.

Manual Controls

Manual controls enable the user to manually rotate a single variable into or out of a projection. This gives fine tuning control to the analyst. Cook & Buja (1997) contains details on the manual controls algorithm. It is similar to a method called spiders proposed by Duffin & Barrett (1994).

Figure 10 illustrates the use of manual controls to examine the results of the LDA index (top left plot, also shown at bottom left in Figure 9). In this result there are three very clearly separated clusters of points. The projection is mostly PC1 (a large positive coefficient), with smaller coefficients for PC2 and PC6. The remaining PCs have effectively zero coefficients. We explore how important these small coefficients are for the three cluster separation. From the optimal projection given by the LDA index we manually rotate PC6 from the projection, and follow by rotating PC2 from the projection:

$$
\mathbf{A} = \begin{bmatrix} 0.889 \\ 0.435 \\ 0.040 \\ 0.053 \\ 0.033 \\ 0.122 \end{bmatrix} \implies \begin{bmatrix} 0.896 \\ 0.439 \\ 0.040 \\ 0.053 \\ 0.033 \\ 0.004 \end{bmatrix} \implies \begin{bmatrix} 0.938 \\ 0.339 \\ 0.042 \\ 0.055 \\ 0.035 \\ 0.004 \end{bmatrix} \implies \begin{bmatrix} 0.996 \\ 0.026 \\ 0.045 \\ 0.059 \\ 0.037 \\ 0.004 \end{bmatrix}
$$

Using manual controls PC6 is rotated out of the projection (top right). (Note that all the coefficients change because they are constrained by orthonormality of the p-dimensional data frame. But notice that the coefficient for PC6 is reduced to effectively zero.) There is very little change to the projected data, so this variable might be ignored. Next we explore the importance of PC2, by rotating it out of the projection. A small change in the coefficient for PC2 results in a blurring of the gap between the two leftmost clusters (bottom left plot). When PC2 is completely removed (bottom right plot) the two leftmost clusters are indistinguishable. But the right cluster is still separated. This suggests that PC2 is important for separating the two leftmost clusters but not important for separating the right cluster.

Precomputed Choices

One of the simplest choices of target planes, which creates a smooth transition from scatterplot matrices is the little tour (McDonald 1982) which interpolates between the frames of a scatterplot matrix, all pairwise marginal projections. Conway, Hardin & Sloane (1996) proposes a method for choosing a fixed number of target planes that are approximately equi-spaced. It chooses the target planes using packing methods on polytopes, and determines a shortest (Hamiltonian) path through the set of targets. Neither of these methods is implemented in GGobi.
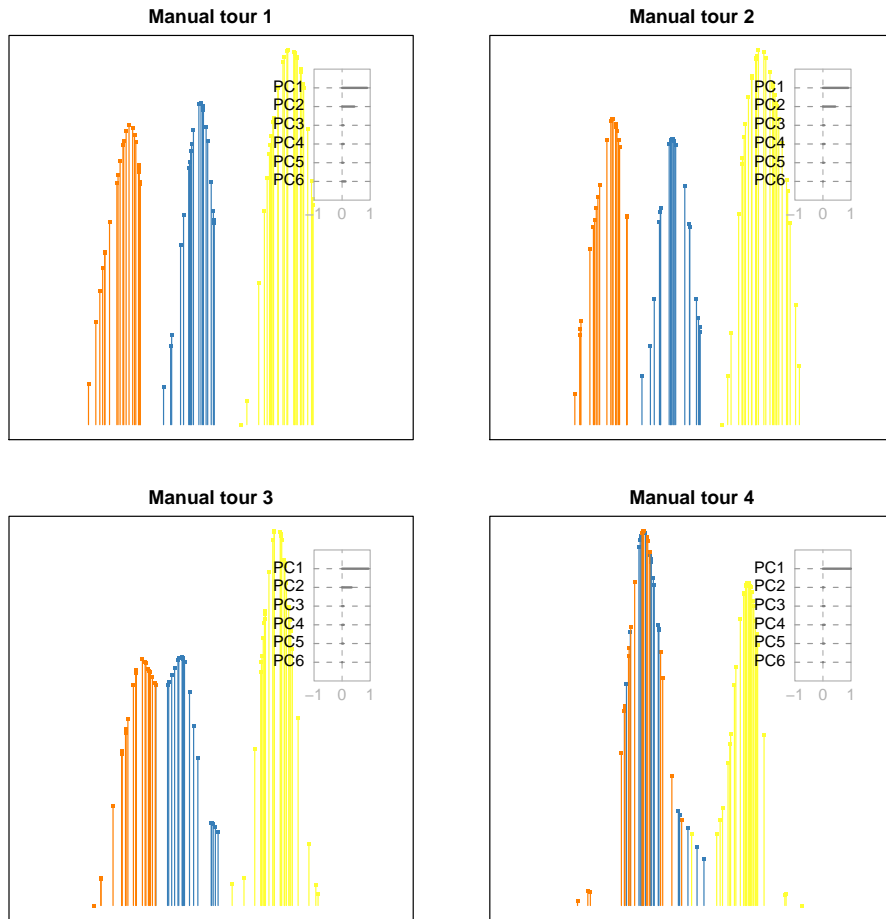
Fig. 10. Manual controls used to examine the results of the LDA index. The sensitivity of the structure to variables PC6 and PC2 is explored.

2.4 A Note on Transformations

When analyzing data it is common to transform variables to examine them on different scales. Transformation plays a useful role prior to PP, also. The most common transformation before PP is to "sphere" the data. Sphering the data means to compute the principal components of the data and to use the resulting variables instead of the original variables. The major reason to do this is that covariance structure is not interesting to find using PP. Covariance can be adequately captured by principal component analysis. Consequently we commonly remove the covariance from the data before running PP, and search for other types of structure in the data. In Figure 9 the labels of the

variables in some of the plots are $PC1, PC2, ...$ reflecting that the data was sphered prior to running the PP guided tour.

2.5 A Note on Scaling

Plots of data are generally constructed by scaling the data using the minimum and maximum data values to fit the data into a plotting space, on a computer screen window, or page of paper. Axes are provided so the viewer can convert the points into the original scales.

For high-dimensional data each variable is scaled to a uniform scale, say $(-1, 1)$, using the minimum and maximum, fitting the data into a p-dimensional hyperrectangle. This scaled data is projected into a plotting space. Its interesting to think about scaling the data after a projection is computed. The effect of this would be a discontinuity between projection frames, like a camera lens constantly zooming and panning as a scene changes.

The PP guided tour operates on the unscaled data values. (It may also be important to transform the data by standardizing variables or sphering before running PP, as discussed in the previous paragraph.) The process of scaling data into a plotting space is called the data pipeline, and is discussed in detail in Buja, Asimov, Hurley & McDonald (1988), Sutherland, Rossini, Lumley, Lewin-Koh, Dickerson, Cox & Cook (2000), and in a different sense in Wilkinson (1999), and Pastizzo, Erbacher & Feldman (2002).

3 Using Tours with Numerical Methods

Tours are useful when used along with numerical methods for certain data analyses, such as dimension reduction and supervised and unsupervised classifcation. We'll demonstrate with an example from supervised classification.

In supervised classification we seek to find a rule for classifying new observations into classes based on a known classes and measured variables in a training set. There are many numerical methods that tackle this problem.

For the data set shown in Figure 11 there are eight variables, and three known classes. A classification tree chooses just two of the variables, eicosenoic and linoleic, to separate the three classes. For the training sample, eicosenoic separates one class (plotted as circles) from the other two, and linoleic separates the remaining two classes (plusses and triangles). The separation of these last two groups although difficult to see in the plot of eicosenoic against linoleic is real (scatterplot matrix at left). There is no gap between the groups of points but it is possible to draw a line with one points from one class on one side, and the points from the other class on the other side. By using a tour we would have noticed that there is a big gap between the three classes using all eight variables, and that also that choosing just three provides a very neat separation. It would be difficult to guess from pairwise plots that arachidic has

an important role but from the tour we can see that when arachidic is combined with linoleic the two classes are much better separated (right plot). The tour projection shows the combination of linoleic and arachidic plotted horizontally which reveals the gap. The tree solution was simple but inadequate, and a small change to the solution provides a much better result.
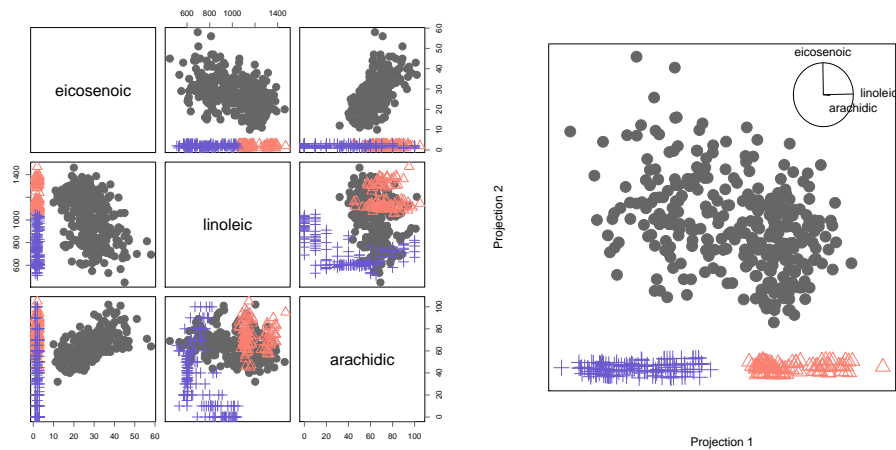

Fig. 11. (Left plot) Scatterplot matrix of three of the important variables for separating the three classes. A single classification tree usually produces the result to split the three classes based on two variables linoleic and eicosenoic. (Right) A projection of linoleic and arachidic, along with eicosenoic, produces a better separation between the classes.

The tree algorithm was hampered by both variable-wise operation and greediness. It doesn't see the combination of linoleic and arachidic because it can only use one variable at each step. It also stops immediately when a separation between the classes was found, having no sense of a bigger gap elsewhere. All numerical methods have assumptions or algorithm constraints or complexity which set limits on the results. A classical method such as linear discriminant analysis assumes that the classes in the data arise from a mixture of normal distributions having equal variance-covariance. Linear discriminant analysis finds a best separating projection similar to the tree solution, one group is well-separated and the other two groups slightly overlapped. It is blinded by the assumption that the three classes have equal variance-covariance. Quadratic discriminant analysis does better in making a rule but cannot provide a good picture of the solution. Black box methods such as forests and neural networks are generally difficult to tease through to understand the solution, but by mapping out a picture of the class structure of this data using a tour we can better understand how they have worked and the resulting solution.

Tours can help numerical approaches in many ways: to choose which of the tools at hand work best for a particular problem, to understand how the tools work in a particular problem, and to overcome the limitations of a particular tool to improve the solution.

4 End Notes

There are numerous recent development in tours that should be noted. Huh & Kim (2002) describes a grand tour with trailing tails marking the movement of the points in previous projections. Tours can be constructed in different projection dimensions and constraints. Yang (1999) describes a grand tour with 3D data projections in virtual environments. The correlation tour described by Buja et al. (1986), and available in GGobi, runs two independent tours of 1D projections on horizontal and vertical axes. This paper also describes constraining the tour to special subspaces such as principal components or canonical coordinates. XGobi (Swayne, Cook & Buja 1998) contained tools for freezing some axes and touring in the constrained complement space, and also a section tour, where points outside a fixed distance from the projection plane were erased. Wegman, Poston & Solka (1998) and Symanzik, Wegman, Braverman & Luo (2002) discuss a tour on the multivariate measurements constrained on spatial locations, which is similar to the multivariate time series and spatial tours where 1D projections are shown against a time variable, or two spatial locations, discussed Sutherland et al. (2000).

In summary, tours support exploring real-valued data. They deliver many projections of real-valued data in an organized manner, allowing the viewer to see the data from many sides.

References

Asimov, D. (1985), 'The Grand Tour: A Tool for Viewing Multidimensional Data', *SIAM Journal of Scientific and Statistical Computing* **6**(1), 128–143.

Asimov, D. & Buja, A. (1994), The Grand Tour via Geodesic Interpolation of 2-Frames, *in* 'Visual Data Exploration and Analysis, Symposium on Electronic Imaging Science and Technology', IS&T/SPIE (Soc. Imaging Sci. Technology/Internat. Soc. Optical Engrg.

Becker, R. A. & Cleveland, W. S. (1987), Brushing Scatterplots, *in* Cleveland & McGill (1988), pp. 201–224.

Buja, A. & Asimov, D. (1986), 'Grand Tour Methods: An Outline', *Computing Science and Statistics* **17**, 63–67.

Buja, A., Asimov, D., Hurley, C. & McDonald, J. A. (1988), Elements of a Viewing Pipeline for Data Analysis, *in* Cleveland & McGill (1988), pp. 277–308.

Buja, A., Cook, D., Asimov, D. & Hurley, C. (2005), Computational Methods for High-Dimensional Rotations in Data Visualization, *in* C. R. Rao, E. J. Wegman & J. L. Solka, eds, 'Handbook of Statistics: Data Mining and Visualization', Elsevier/North Holland, `http://www.elsevier.com`, pp. 391–413.

Buja, A., Hurley, C. & McDonald, J. A. (1986), A Data Viewer for Multivariate Data, *in* T. J. Boardman & I. M. Stefanski, eds, 'Proceedings of the 18th Symposium on the Interface between Comput. Sci. and Statist.', Elsevier, Fort Collins, CO, pp. 171–174.

Carr, D. B., Wegman, E. J. & Luo, Q. (1996), ExplorN: Design Considerations Past and Present, Technical Report 129, Center for Computational Statistics, George Mason University.

Cleveland, W. S. & McGill, M. E., eds (1988), *Dynamic Graphics for Statistics*, Wadsworth, Monterey, CA.

Conway, J. H., Hardin, R. H. & Sloane, N. J. A. (1996), 'Packing Lines, Planes etc., Packings in Grassmannian Spaces', *Experimental Mathematics* **5**, 139–159.

Cook, D. & Buja, A. (1997), 'Manual Controls For High-Dimensional Data Projections', *Journal of Computational and Graphical Statistics* **6**(4), 464–480. Also see `www.public.iastate.edu/~dicook/research/papers/manip.html`.

Cook, D., Buja, A., Cabrera, J. & Hurley, C. (1995), 'Grand Tour and Projection Pursuit', *Journal of Computational and Graphical Statistics* **4**(3), 155–172.

Duffin, K. L. & Barrett, W. A. (1994), Spiders: A New Interface for Rotation and Visualization of N-Dimensional Point Sets, *in* 'Proceedings of Visualization '94', IEEE Computer Society Press, Los Alamitos, CA, pp. 205–211.

Huber, P. J. (1985), 'Projection Pursuit (with discussion)', *Annals of Statistics* **13**, 435–525.

Huh, M. Y. & Kim, K. (2002), 'Visualization of Multidimensional Data Using Modifications of the Grand Tour', *Journal of Applied Statistics* **29**(5), 721–728.

Johnson, R. A. & Wichern, D. W. (2002), *Applied Multivariate Statistical Analysis (5th ed)*, Prentice-Hall, Englewood Cliffs, NJ.

Lee, E. K., Cook, D., Klinke, S. & Lumley, T. (2005), 'Projection pursuit for exploratory supervised classification', *Journal of Computational and Graphical Statistics* p. To appear.

McDonald, J. A. (1982), Interactive Graphics for Data Analysis, Technical Report Orion II, Statistics Department, Stanford University.

Pastizzo, M. J., Erbacher, R. F. & Feldman, L. B. (2002), 'Multi-dimensional data visualization', *Behavior Research Methods, Instruments, and Computers* **34**(2), 158–162.

Posse, C. (1990), 'An Effective Two-dimensional Projection Pursuit Algorithm', *Communications in Statistics: Simulation and Computation* **19**, 1143–1164.

Sutherland, P., Rossini, A., Lumley, T., Lewin-Koh, N., Dickerson, J., Cox, Z. & Cook, D. (2000), 'Orca: A Visualization Toolkit for High-Dimensional Data', *Journal of Computational and Graphical Statistics* **9**(3), 509–529.

Swayne, D. F., Cook, D. & Buja, A. (1998), 'XGobi: Interactive Dynamic Graphics in the X Window System', *Journal of Computational and Graphical Statistics* **7**(1), 113–130. See also `www.research.att.com/areas/stat/xgobi/`.

Swayne, D., Temple Lang, D., Cook, D. & Buja, A. (2001-), 'GGobi: software for exploratory graphical analysis of high-dimensional data', *[Jul/01] Available publicly from* `www.ggobi.org`.

Symanzik, J., Wegman, E. J., Braverman, A. & Luo, Q. (2002), 'New applications of the image grand tour', *Computing Science and Statistics* **34**, 500–512.

Tierney, L. (1991), *LispStat: An Object-Orientated Environment for Statistical Computing and Dynamic Graphics*, Wiley, New York, NY.

Wegman, E. J. (1991), 'The Grand Tour in k-Dimensions', *Computing Science and Statistics* **22**, 127–136.

Wegman, E. J., Poston, W. L. & Solka, J. L. (1998), Image grand tour, *in* 'Automatic Target Recognition VIII - Proceedings of SPIE, 3371', pp. 286–294.

Wilkinson, L. (1999), *The Grammar of Graphics*, Springer, New York, NY.

Yang, L. (1999), 3D Grand Tour for Multidimensional Data and Clusters, *in* 'Lecture Notes In Computer Science, Proceedings of the Third International Symposium on Advances in Intelligent Data Analysis', Vol. 1642, Springer-Verlag, London, UK, pp. 173–186.