Grand Tours, Projection Pursuit Guided Tours, and Manual Controls

III.2

Dianne Cook, Andreas Buja, Eun-Kyung Lee, Hadley Wickham

2.1 Introductory Notes 296
Some Basics on Projections 297
What Structure Is Interesting? 299
2.2 Tours 301
Terminology: Plane, Basis, Frame, Projection 302
Interpolating Between Projections: Making a Movie 302
Choosing the Target Plane 303
A Note on Transformations 310
A Note on Scaling 310
2.3 Using Tours with Numerical Methods 310
2.4 End Notes 312

2.1 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 analysis (PCA) and linear discriminant analysis. PCA (e.g., Johnson and Wichern 2002) chooses a projection to maximize the variance. Fisher's linear discriminant (e.g., Johnson and Wichern 2002) chooses a projection that maximizes the relative separation between group means. Projection pursuit (PP) (e.g., Huber 1985) generalizes these ideas into a common strategy, where an arbitrary function on projections is optimized. The scatterplot matrix (e.g., Becker and Cleveland 1987) also can be considered to be a projection method. It shows projections of the data onto all pairs of coordinate axes, the 2-D marginal projections of the data. These projection methods choose a few select projections out of infinitely many.

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, until a wise man settled the fight. They were all correct, but each described 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: walk randomly to discover unexpected sights, employ a guide, or guide ourselves using a map. These modes of exploration are matched by three commonly available types of tours. They are the tours available in the software, GGobi (Swayne et al., 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 *PP 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 PP 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 projec-

tion 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.

Some Basics on Projections

2.1.1

What is a projection? We can think of a projection as the shadow of an object. Especially if it is a 2-D projection, then the projection is the shadow the object casts under a bright light (Fig. 2.1). If the object rotates in the light, we see many different 2-D shadows and we can infer the shape of the object itself.

Figure 2.1. Projections are like shadows. When many projections are viewed, it is possible to obtain a sense of the shape of a dataset. What may look like a horse in one projection may be revealed as a carefully oriented pair of hands by another

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

Here is a concrete example. Suppose our data matrix and projection were these:

$$\mathbf{X} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 15 \\ 0 & 15 & 0 \\ 0 & 15 & 15 \\ 15 & 0 & 0 \\ 15 & 0 & 15 \\ 15 & 15 & 0 \\ 15 & 15 & 15 \end{bmatrix}_{8\times 3} \quad \text{and} \quad \mathbf{A}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}_{3\times 2} \quad \text{then} \quad \mathbf{XA}_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 15 \\ 0 & 15 \\ 15 & 0 \\ 15 & 0 \\ 15 & 15 \\ 15 & 15 \end{bmatrix}_{8\times 2}$$

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

$$\mathbf{A}_2 = \begin{bmatrix} 0.71 & -0.42 \\ 0.71 & 0.42 \\ 0 & 0.84 \end{bmatrix}_{3\times 2} \quad \text{then} \quad \mathbf{XA}_2 = \begin{bmatrix} 0 & 0 \\ 0 & 12.60 \\ 10.65 & 6.30 \\ 10.65 & 18.90 \\ 10.65 & -6.30 \\ 10.65 & 6.30 \\ 21.30 & 0 \\ 21.30 & 12.60 \end{bmatrix}_{8\times 2}$$

is a combination of all three variables.

These projections are illustrated in Fig. 2.2. The top row shows the data projections, $\mathbf{XA}_1$ and $\mathbf{XA}_2$, respectively. The bottom row displays the projection coefficients, $\mathbf{A}_1$ and $\mathbf{A}_2$. A row in $\mathbf{A}$ can also be interpreted as the projection of the coordinate axis (p-dimensional to d-dimensional) for each variable, and it is represented by a line in this display. The length and direction of the line displays the contribution each variable makes to the projected data view. In $\mathbf{A}_1$ the data projection is constructed purely from 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 variable dimensions of the data and, hence, enable to the analyst to interpret the data.

We also commonly use 1-D projections in data analysis. With a 1-D projection we typically use a histogram or density plot to display the data. Consider the 2-D data in Fig. 2.3 (left plot) and two 1-D 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.

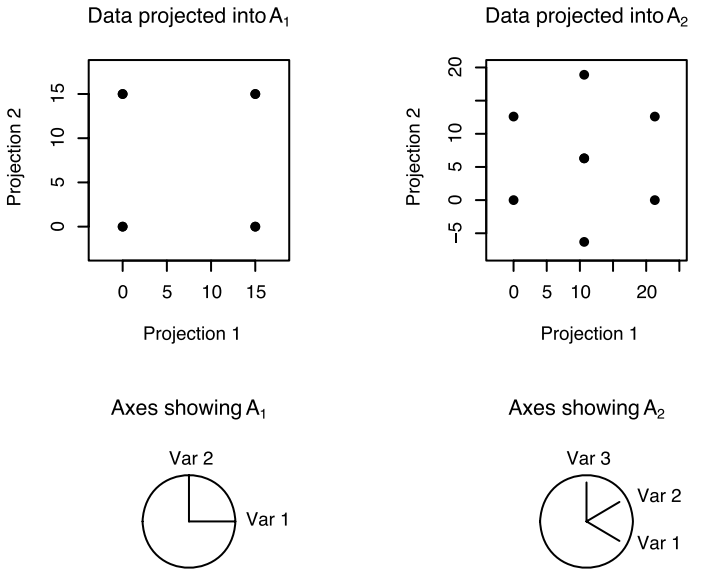

Figure 2.2. Two 2-D data projections. The two *top plots* are the data projections, and the two *bottom plots* are illustrations of the projection coefficients

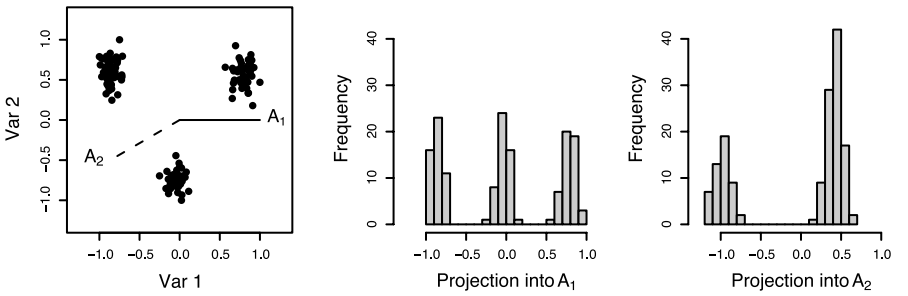

Figure 2.3. Two 1-D projections of 2-D data

What Structure Is Interesting? 2.1.2

When we use tours, what are we looking for in the data? We search for data projections that are *not* bell-shaped and, hence, not normally distributed, for example, clusters of points, outliers, nonlinear relationships, and low-dimensional substructures. All of these can be present in multivariate data but hidden from the viewer who only chooses a few static projections. Figures 2.4 and 2.5 show some examples.

In Fig. 2.4 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

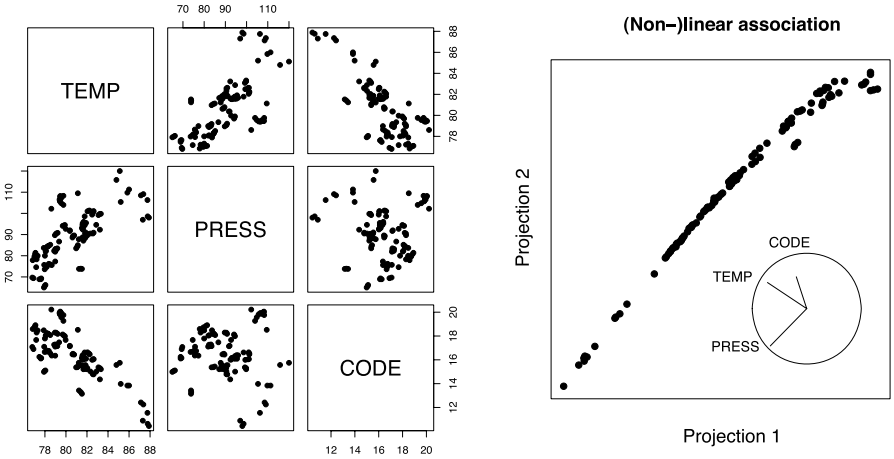

Figure 2.4. The three variables show some association in the scatterplot matrix (all pairwise marginal projections in *left plot*), but they are revealed to be almost perfectly related by a tour (*right plot*)

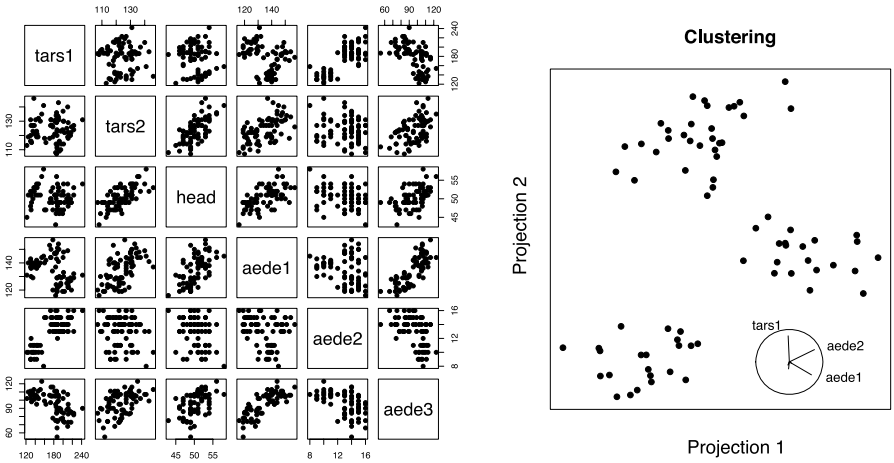

Figure 2.5. The six variables show some clustering in the scatterplot matrix (all pairwise marginal projections in *left plot*), but the three clusters are much better separated in a tour projection (*right plot*)

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:

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

In Fig. 2.5 (left) the pairwise scatterplots suggest there is some clustering of the data points in this six-variable dataset. The tour projection (right) reveals three well-separated clusters. The projection revealing the clusters is:

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

which is primarily a combination of three of the six variables: tars1, aede1, aede2.

Tours

2.2

Most of us are familiar with 3-D rotation, which is something we can do in the real world. We can take an object and rotate it with our hands 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 et al., 1996; Tierney, 1991) or by moving a $d(< p)$-dimensional plane through the space and projecting the data onto it. The latter approach is like looking at the data from different sides.

Movement of a projection plane is achieved by selecting a starting plane and a target plane and computing intermediate planes using a geodesic interpolation. A geodesic is a circular path, which is generated by constraining the planar interpolation to produce orthonormal descriptive frames. 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 by interpolating from plane to plane, rather than frame to frame. 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 as in Fig. 2.6. This type of within-plane rotation is distracting to the viewer, akin to viewing a scene while standing on

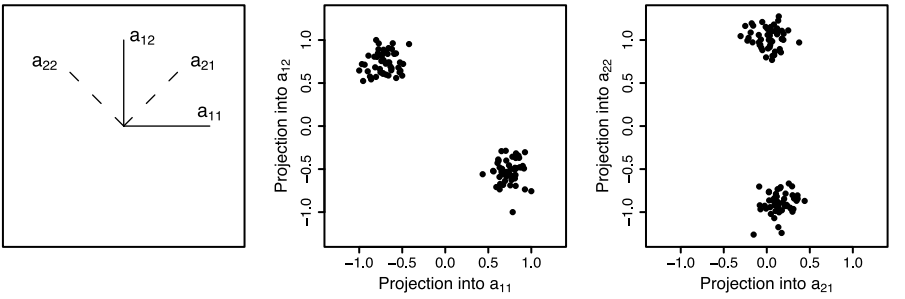

Figure 2.6. Two different frames that describe the same plane, and the resulting rotated views of the data

a wobbly platform. Planar rotations are discussed in detail in Asimov (1985), more simply in Buja and Aasimov (1986a), and very technically in Buja et al. (2005), and in Asimov and Buja (1994), and Buja et al. (1986b) as well.

Differences in the method of selecting the target plane yield different types of tours. The grand tour uses a random selection of target plane. The guided tour quantifies the structure present in each projection and uses this to guide the choice of target plane. Manual controls let the user choose the target direction by manipulating the values of projection coefficients.

2.2.1 Terminology: Plane, Basis, Frame, Projection

It is conventional to use a p-dimensional orthonormal basis:

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

to describe p-dimensional Euclidean space. A d-dimensional plane in p-space can be defined by an infinite number of d-dimensional orthonormal frames. For example, consider the $d = 2$-dimensional frames:

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

both of which describe the same 2-D plane. We conventionally use $\mathbf{A}_1$ as the frame describing the 2-D plane, but we could just as validly use $\mathbf{A}_2$. Figure 2.6 illustrates the two frames, which result in the same but rotated projections of the data.

In GGobi tours, we generate a new target basis and use this to define the target plane. But the actual basis used to create the data projection is a within-plane rotation of the target basis that matches the basis describing the starting plane.

2.2.2 Interpolating Between Projections: Making a Movie

A movie of data projections is created by interpolating along a geodesic path from the current (starting) plane to the new target plane. The algorithm 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. It is important to check that $\mathbf{A}_a$ and $\mathbf{A}_z$ describe different planes, and generate a new $\mathbf{A}_z$ if not. To find the optimal rotation of the starting plane into the target plane we need to find the frames in each plane that are the closest.

2. Determine the shortest path between frames using singular value decomposition. $\mathbf{A}_a'\mathbf{A}_z = \mathbf{V}_a\Lambda\mathbf{V}_z'$, $\Lambda = \text{diag}(\lambda_1 \geq \cdots \geq \lambda_d)$, and the principal directions in each plane are $\mathbf{B}_a = \mathbf{A}_a\mathbf{V}_a$, $\mathbf{B}_z = \mathbf{A}_z\mathbf{V}_z$, a within-plane rotation of the descriptive bases $\mathbf{A}_a, \mathbf{A}_z$, respectively. The principal directions are the frames describing the starting and target planes that 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.

Choosing the Target Plane 2.2.3

Grand Tour

The grand tour method for choosing the target plane is to use random selection. A frame is randomly selected from the space of all possible projections.

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 2-D projection, where the second vector is orthonormalized on the first.

Figure 2.7 illustrates the tour path, using GGobi to look at itself. Using GGobi, we recorded the sequence of 9000 1D projections displayed of 3-D data. This tour path is a set of 9000 points on a 3-D sphere, where each point corresponds to a projection. We use a tour to view the path (top left plot). The starting projection is $\mathbf{A}' = (1\ 0\ 0)$, indicated by a large point (●) or solid circle in the display. It is at the center right in the plot, a projection in the first two variables. The corresponding data projection is shown at top right. The grand tour path zigzags around the 3-D sphere. The grand tour can be considered as an interpolated random walk over the space of all planes. With enough time it will entirely cover the surface of the sphere. The bottom row of plots shows two views of a grand tour path of 20 000 1-D projections of 6-dimensional data.

Projection Pursuit Guided Tour

In a guided tour (Cook et al., 1995) the next target basis is selected by optimizing a PP index function. The index function numerically describes what is interesting

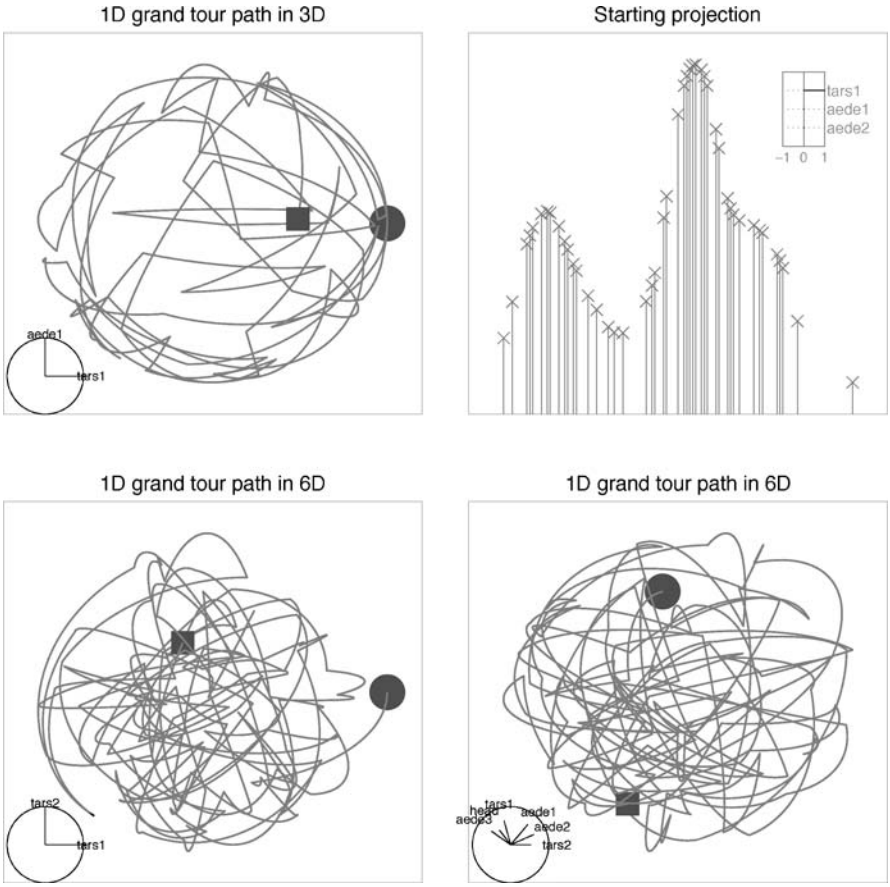

Figure 2.7. Some views of 1-D grand tour paths in 3 dimensions (*top left*) and 6 dimensions (*bottom*). The path consists of a sequence of points on a 2-D and 6-D sphere respectively. Each *point* corresponds to a projection from 3 dimensions (or 6 dimensions) to 1 dimension. The *solid circle* indicates the first point on the tour path corresponding to the starting frame, yielding the 1-D data projection (*top right*) shown for the 3-D path. The *solid square* indicates the last point in the tour path, or the last projection computed

in a projection: higher values correspond to more interesting structure in the projections. Used alone, PP seeks out low-dimensional projections that expose interesting features of the high-dimensional point cloud. In conjunction with the interpolation, a PP guided tour shows many projections to the viewer, in a smooth sequence. Using a PP index function to navigate the high-dimensional data space has the advantage over the grand tour of increasing the proportion of interesting projections visited.

The PP index, $f(\mathbf{XA})$, is optimized over all possible d-dimensional projections of p-dimensional data, subject to orthonormality constraints on $\mathbf{A}$. The optimization

procedure is an important part of a PP guided tour. The purpose of PP optimization is to find all of the interesting projections, so an optimization procedure needs to be flexible enough to find global and local maxima. It should not doggedly search for a global maximum, but it should spend some time visiting local maxima.

Posse (1990) compared several optimization procedures and suggest a random search for finding the global maximum of a PP index. Cook et al. (1995) used a derivative-based optimization, always climbing the nearest hill, which when merged with a grand tour was a lot like interactive simulated annealing. Klein and Dubes (1989) showed that simulated annealing can produce good results for PP.

Lee et al. (2005) use the modified simulated annealing method. It uses two different temperatures, one for neighborhood definition and the other (cooling parameter) for the probability that guards against getting trapped in a local maximum. This allows the algorithm to visit a local maximum and then jump out and look for other maxima. The temperature of the neighborhood is rescaled by the cooling parameter enabling escape from the local maximum. The optimization algorithm used in GGobi follows these steps:

1. From the current projection, $\mathbf{A}_a$, calculate the initial PP index value, $I_0 = f(\mathbf{X}\mathbf{A}_a)$.
2. Generate new projections, $\mathbf{A}_i^* = \mathbf{A}_a + c\mathbf{A}_i$, from a neighborhood of the current projection where the size of the neighborhood is specified by the cooling parameter, c, and $\mathbf{A}_i$ is a random projection.
3. Calculate the index value for each new projection, $I_i = f(\mathbf{X}\mathbf{A}_i^*)$.
4. Set the projection with the highest index value to be the new target, $\mathbf{A}_z = \mathbf{A}_{\max_i I_i}$, and interpolate from $\mathbf{A}_a$ to $\mathbf{A}_z$.

Figure 2.8 (top two plots) shows a PP guided tour path (1-D in three dimensions). 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 negative counterpart (both highlighted by large solid black circles). 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 by large solid black circles) are the maxima of the PP index, and for the most part, these are at the same projection. The corresponding data projections (approximately 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 the formula for holes, central Mass, and PCA indices, it is assumed that $\mathbf{X}$ is sphered using PCA, that is, the mean is zero and the variance–covariance is equal to the identity matrix. This assumption is not necessary for the LDA index.

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

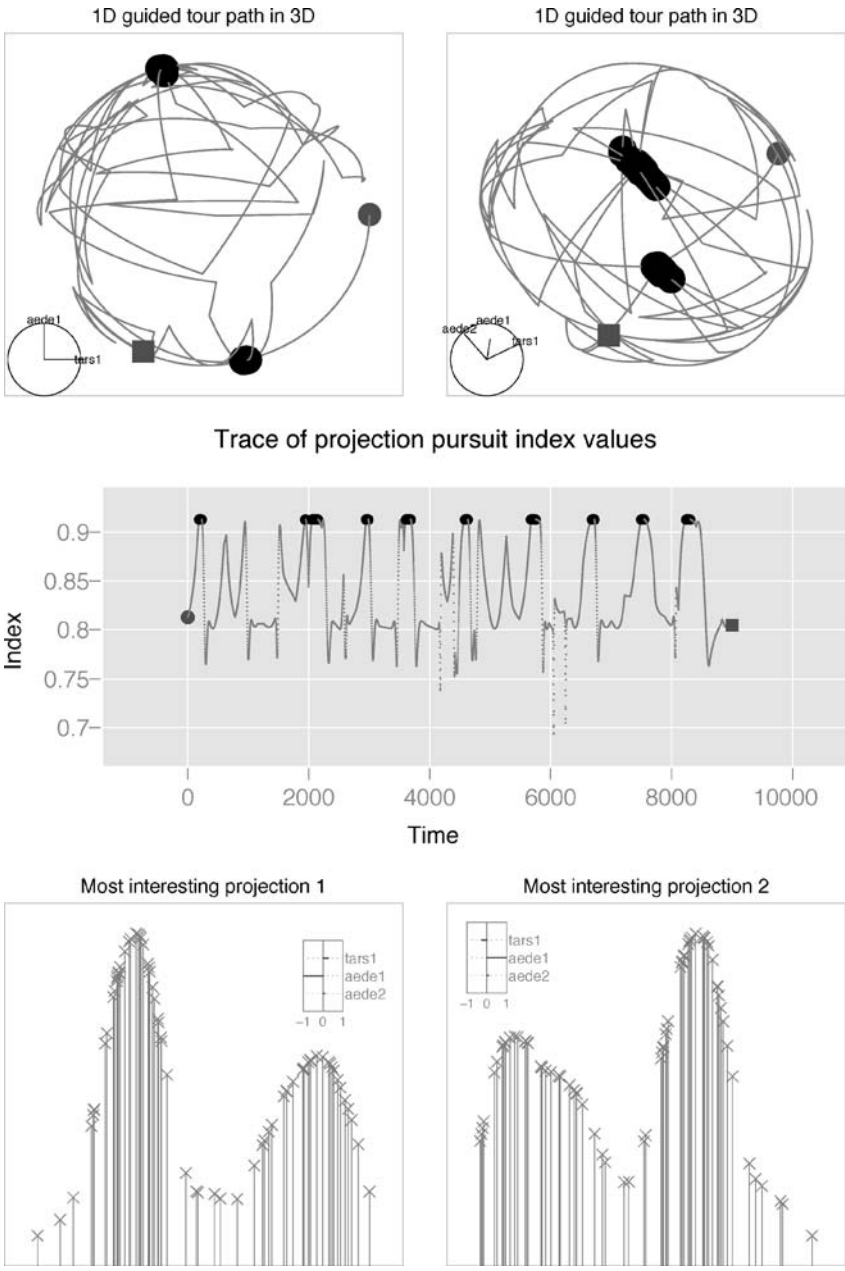

Figure 2.8. Projection pursuit guided tours. *Top*: path of 1-D projections in 3 dimensions. *Middle*: time trace of PP index. *Bottom*: data projections corresponding to PP index maxima. The maxima are highlighted by large *solid circles*. It is interesting here that the optimization keeps returning the tour to similar neighborhoods in the projection space, corresponding to bimodal densities in the 1-D projections

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 $\mathbf{XA} = \mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n]^T$ is an $n \times d$ 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, $n_i, i = 1, \ldots, g$ is the number of cases in each group.

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

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

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

Figure 2.9 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 trimodal 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 and Buja (1997) has details on the manual controls algorithm. It is similar to a method called spiders proposed by Duffin and Barrett (1994).

Figure 2.10 illustrates the use of manual controls to examine the results of the LDA index (top left plot, also shown at bottom left in Fig. 2.9). In this view 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 the importance of these small coefficients for the three-cluster separation. From the optimal projection given by the LDA index we manually rotate PC6 out of the projection and follow by rotating PC2 out of the projection:

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

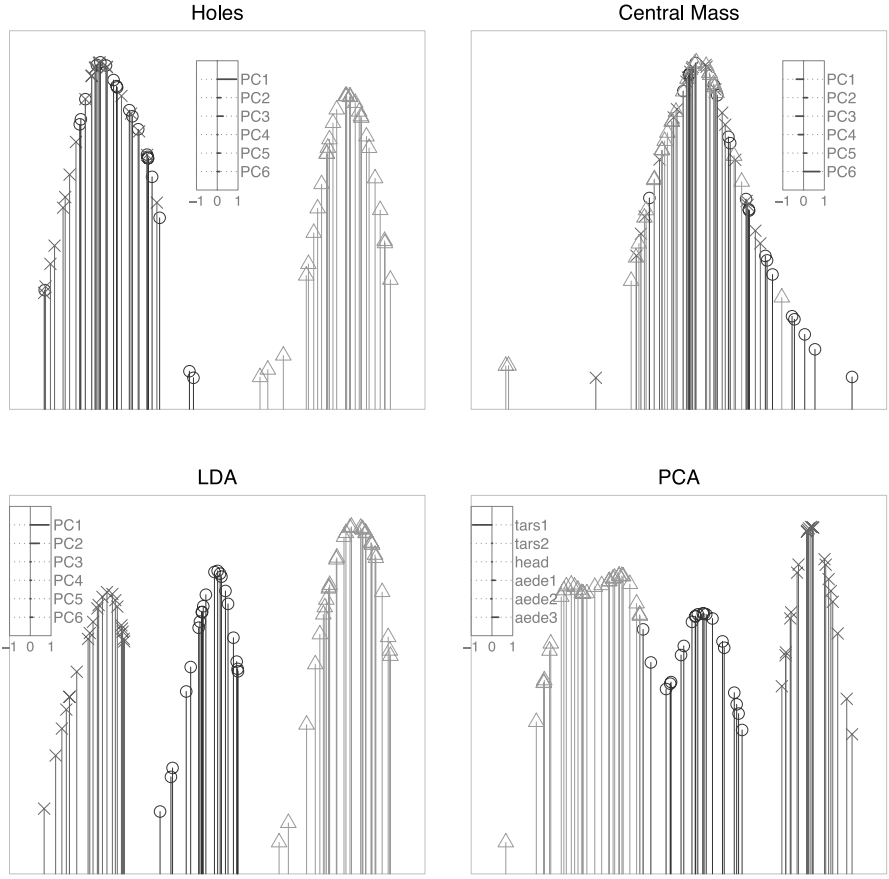

Figure 2.9. One-dimensional data projections corresponding to maxima from four different PP indices computed on the same data. The interesting feature of the data is the separation of the three classes. *Top left*: the holes index finds a projection with a hole in the middle, where one cluster is separated from the other two. *Top right*: the central mass index finds a projection where most points are clumped in the center, revealing a few outliers. *Bottom left*: LDA, using the class information, finds a separation of all three clusters. *Bottom right*: the PCA index finds a projection where all three classes are somewhat distinct

PC6 is rotated out of the projection first (Fig. 2.10, top right). Note that all the coefficients change some because they are constrained by the orthonormality of the p-dimensional data frame. But notice that the coefficient for PC6 is effectively reduced to 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 (Fig. 2.10, bottom row). 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.

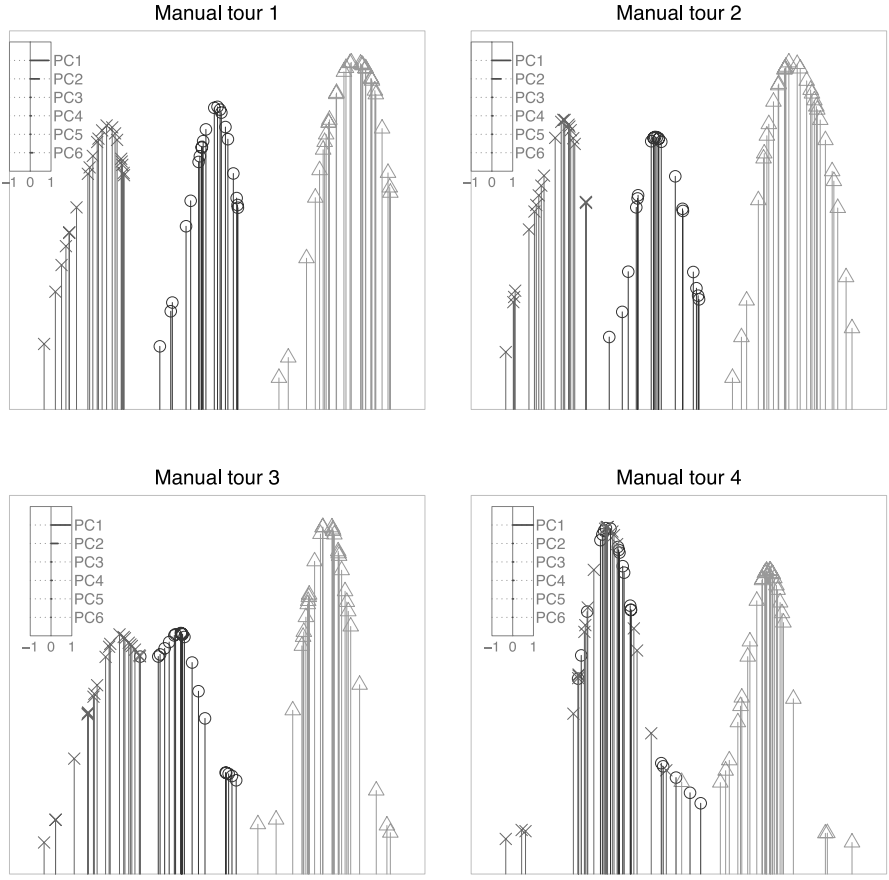

Figure 2.10. Manual controls used to examine the sensitivity of the clustering revealed by the LDA index to PC6 and PC2 is explored. *Top left to top right plots*: coefficient on PC6 reduced from 0.122 to 0.004 gives a smaller gap between clusters, but the clusters are still separable. *Bottom left to bottom right*: coefficient for PC2 reduced from 0.439 to 0.026 which removes the cluster structure

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 that creates a smooth transition from scatterplot matrices is the little tour (McDonald, 1982) that interpolates between the frames of a scatterplot matrix of all pairwise marginal projections. Conway et al. (1996) proposes a method for choosing a fixed number of target planes that are approximately equispaced. 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.

2.2.4 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 as well. 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 we are not interested in covariance structure. This is adequately captured by PCA. Consequently we commonly remove the covariance from the data before running PP and search for other types of structure in the data. In Fig. 2.9 the labels of the variables in some of the plots $PC1, PC2, \ldots$ reflect that the data were sphered prior to running the PP guided tour. Sometimes transformations are performed to relieve the data of outliers or skewness. When these occur in single variables, they can be detected and addressed before running PP, but PP is useful for detecting multivariate outliers and nonlinear dependencies in high-dimensional data.

2.2.5 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 sheet 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 using the minimum and maximum, packing the data into a p-dimensional hyperrectangle. These scaled data are projected into a plotting space. It might interesting to think about scaling the data after a projection is computed, but the effect of this approach is a discontinuity from one projection frame to the next. It would be like watching a movie where the camera lens constantly zooms and pans.

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 et al. (1988), Sutherland et al. (2000), and in a different sense, in Wilkinson (1999) and Pastizzo et al. (2002).

2.3 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 classification. We'll demonstrate with an example from supervised classification.

In supervised classification we seek to find a rule for predicting the class of new observations based on training a classifier using known classes. There are many numerical methods that tackle this problem.

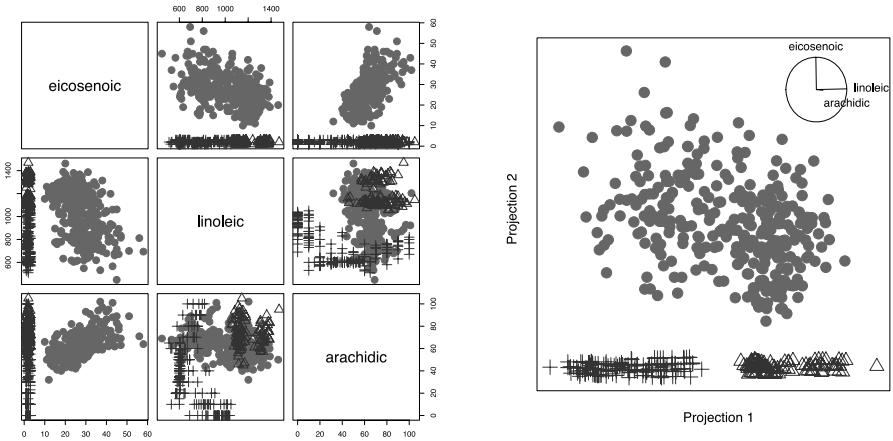

Figure 2.11. *Left plot*: scatterplot matrix of three of the important variables for separating the three classes. A single classification tree usually produces the result to split the three classes based on two variables, linoleic and eicosenoic. *Right*: a projection of linoleic and arachidic, along with eicosenoic, produces a better gap between the classes

For the dataset shown in Fig. 2.11 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 points from one class on one side of it 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 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 that reveals the gap. The tree solution was simple but inadequate, and a small change to the solution provides a much better result.

The tree algorithm was hampered by both variable wise operation and greediness. It did not see the combination of linoleic and arachidic because it could only use one variable at each step. It also stopped 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 that sets 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 over-

lapped. 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. The solution of black-box methods such as forests and neural networks are generally difficult to understand, but by mapping out a picture of the class structure of these 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 works 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.

2.4 End Notes

There are numerous recent developments in tours that should be noted. Huh and Kim (2002) describes a grand tour with trailing tails marking the movement of the points in previous projections. The tour can be constructed in different projection dimensions and constraints. Yang (1999) describes a grand tour with 3-D data projections in virtual environments. The correlation tour described by Buja et al. (1986b), and available in GGobi, runs two independent tours of 1-D 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 et al., 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 place were erased. Wegman et al. (1998) and Symanzik et al. (2002) discuss a tour on the multivariate measurements constrained on spatial locations, which is similar to the multivariate time series tour discussed in Sutherland et al. (2000), where 1-D projections are shown against a time variable.

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 J Sci Stat Comput* 6(1):128–143.

Asimov, D. and Buja, A. (1994). The grand tour via geodesic interpolation of 2-Frames, *Visual Data Exploration and Analysis, Symposium on Electronic Imaging Science and Technology*, IS&T/SPIE.

Buja, A. and Asimov, D. (1986a). Grand tour methods: an outline, *Comput Sci Stat* 17:63–67.

Buja, A., Asimov, D., Hurley, H. and McDonald, J.A. (1988). Elements of a Viewing Pipeline for Data Analysis, *in* Cleveland, W.S. and McGill, M.E. (ed), *Dynamic Graphics for Statistics*, Wadsworth, Monterey, CA, pp. 277–308.

Buja, A., Cook, D., Asimov, D. and Hurley, C. (2005). Computational methods for high-dimensional rotations in data visualization, *in* Solka, J.L., Rao, C.R. and Wegman, E.J. (ed), *Handbook of Statistics: Data Mining and Visualization*, Elsevier/North Holland, http://www.elsevier.com, pp. 391–413.

Buja, A., Hurley, C. and McDonald, J.A. (1986b). A Data Viewer for Multivariate Data, *in* Stefanski, I.M. Boardman, T.J. (ed), *Proceedings of the 18th symposium on the interface between computer science and statistics*, Elsevier, Fort Collins, CO, pp. 171–174.

Carr, D.B., Wegman, E.J. and Luo, Q. (1996). ExplorN: design considerations past and present, *Technical report*, Center for Computational Statistics, George Mason University.

Conway, J.H., Hardin, R.H. and Sloane, N.J.A. (1996). Packing lines, planes etc., packings in Grassmannian spaces, *Exp Math* 5:139–159.

Cook, D., Buja, A. (1997). Manual controls for high-dimensional data projections, *J Comput Graph Stat* 6(4):464–480. Also see www.public.iastate.edu/~dicook/research/papers/manip.html.

Cook, D., Buja, A., Cabrera, J. and Hurley, C. (1995). Grand tour and projection pursuit, *J Comput Graph Stat* 4(3):155–172.

Duffin, K.L., Barrett W.A. (1994). Spiders: a new interface for rotation and visualization of N-dimensional point sets, *Proceedings of Visualization '94*, IEEE Computer Society Press, Los Alamitos, CA, pp. 205–211.

Huh, M.Y. and Kim, K. (2002). Visualization of multidimensional data using modifications of the grand tour, *J Appl Stat* 29(5):721–728.

Klein, R.W. and Dubes, R.C. (1989). Experiments in projection and clustering by simulated annealing, *Pattern Recog* 22(2):213–220.

Lee, E.K., Cook, D., Klinke, S. and Lumley, T. (2005). Projection pursuit for exploratory supervised classification, *J Comput Graph Stat* 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. and Feldman, L.B. (2002). Multi-dimensional data visualization, *Behav Res Methods, Instrum Comput* 34(2):158–162.

Posse, C. (1990). An effective two-dimensional projection pursuit algorithm, *Commun Stat Simulat Comput* 19:1143–1164.

Sutherland, P., Rossini, A., Lumley, T., Lewin-Koh, N., Dickerson, J., Cox, Z. and Cook, D. (2000). Orca: a visualization toolkit for high-dimensional data, *J Comput Graph Stat* 9(3):509–529.

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

Swayne, D.F., Cook, D. and Buja, A. (1998). XGobi: interactive dynamic graphics in the X Window system, *J Comput Graph Stat* 7(1):113–130. See also www.research.att.com/areas/stat/xgobi/.

Symanzik, J., Wegman, E.J., Braverman, A. and Luo, Q. (2002). New applications of the image grand tour, *Comput Sci Stat* 34:500–512.

Tierney, L. (1991). *LispStat: an object-orientated environment for statistical computing and dynamic graphics*, Wiley, New York.

Wegman, E.J. (1991). The grand tour in k-dimensions, *Comput Sci Stat* 22:127–136.

Wegman, E.J., Poston, W.L. and Solka, J.L. (1998). Image grand tour, *Automatic Target Recognition VIII – Proc SPIE, 3371*, pp. 286–294.

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

Yang, L. (1999). 3D grand tour for multidimensional data and clusters. *In*: Proceedings of the Third International Symposium on Advances in Intelligent Data Analysis. *Lecture Notes in Computer Science*, vol. 1642, Springer-Verlag, London, UK, pp. 173–186