Clustering Visualizations of Multidimensional Data

Catherine B. Hurley

Many graphical methods for displaying multivariate data consist of arrangements of multiple displays of one or two variables; scatterplot matrices and parallel coordinates plots are two such methods. In principle these methods generalize to arbitrary numbers of variables but become difficult to interpret for even moderate numbers of variables. This article demonstrates that the impact of high dimensions is much less severe when the component displays are clustered together according to some index of merit. Effectively, this clustering reduces the dimensionality and makes interpretation easier. For scatterplot matrices and parallel coordinates plots clustering of component displays is achieved by finding suitable permutations of the variables. I discuss algorithms based on cluster analysis for finding permutations, and present examples using various indices of merit.

Key Words: Parallel coordinates; Permutation of variables; Projection pursuit; Scatterplot matrices.

1. INTRODUCTION

Datasets of three or more dimensions are notoriously difficult to display on a two-dimensional screen or on a piece of paper. Many graphical methods for displaying multivariate data consist of arrangements of multiple displays of one or two variables—for example, a scatterplot matrix consists of all pairwise scatterplots of two variables arranged in a square matrix, and a parallel coordinates display is a sequence of one-dimensional dotplots where line segments are drawn to connect the dots pertaining to a particular case. While in principle these methods generalize to arbitrary numbers of variables, in practice as the dimensions increase, they become less effective, presenting us with an overwhelming amount of information that is difficult to absorb. Usually, the ordering of the variables in these displays is arbitrary and corresponds to the order in which the variables were listed in the data file. However, the interpretability and effectiveness of visualizations often improve

Catherine B. Hurley is Senior Lecturer, Department of Mathematics, National University of Ireland Maynooth, Co. Kildare, Ireland (E-mail: catherine.hurley@may.ie).

Journal of Computational and Graphical Statistics, Volume 13, Number 4, Pages 788–806
DOI: 10.1198/106186004X12425

dramatically when the variables are reordered in some systematic way.

A scatterplot matrix shows all pairwise scatterplots of p variables, while a parallel coordinate display shows $p - 1$ of the $\binom{p}{2}$ pairwise line plots. Some of these pairwise plots are more interesting or informative than others, and an effective visualization should help us to focus on these. Our basic idea is that each pairwise display (a panel) is awarded a merit score measuring its "interestingness." Then the variables are reordered so that the viewer's attention will be focused on the most interesting panels, which are placed in prominent positions. For the scatterplot matrix, we consider positions close to the diagonal to be the most prominent, while for the parallel coordinate display interesting panels should be among the $p - 1$ visible panels. Suitable merit measures will depend on the context of the data and the type of display, but correlation is often a good starting point. Then the visualizations will help us identify clusters of similar (highly correlated) variables, effectively reducing the dimensionality of the visualization problem.

Ideally, the panel merit scores are combined into an overall merit score for the entire display. We could then find the permutation of the variables maximizing this overall score. A brute-force approach to solving this problem evaluates the criterion on all possible permutations of the variables, but this is slow except for small numbers of variables. Because our goal is effective data visualization, it is probably better to find a good display quickly rather than wait around for a slightly better but optimal display. Therefore, we use a fast ad-hoc algorithm based on cluster analysis (Gruvaeus and Wainer 1972) to come up with suitable permutations of the variables. In our experience the resulting visualizations are often far more effective than those using standard variable order.

The problem of choosing an ordering of variables for displays of multivariate data has received surprisingly little attention in the literature. The work of Bertin is an exception in this regard; ordering variables, cases, and categories in so-called "matrix displays" is a major theme of his work (Bertin 1983).

In multiway trellis displays, Cleveland (1995) ordered categories by their medians, Friendly (1994) ordered categories in a mosaic display by their score on the first correspondence analysis direction, and in both cases ordering clarifies patterns present in the data. Carr and Olsen (1996) stated succinctly that "sorting simplifies" and demonstrated this extremely effectively using a minimal spanning tree-based ordering of row and column variables in a two-way layout. Wegman (1990) sorted observations along one variable at a time to produce a variation on the parallel coordinate display called the "color histogram." The "data image" described by Minnotte and West (1998) is similar to the color histogram, but it orders both cases and variables using the the Gruvaeus and Wainer (1972) algorithm.

More recently, Friendly and Kwan (2003) argued very strongly in favor of ordering information in visual displays of data. Their basic notion is that similar variables, cases, and categories should be positioned adjacently in a graphical display, and they used orderings based on eigen decompositions for this purpose. In a related article, Friendly (2002) examined ways of rendering correlation matrices. He advocates reordering variables so that highly correlated variables are positioned adjacently, and computes an ordering from the angular positions of the first two eigen vectors of the correlation matrix.

In the visualization literature, Ankerst, Berchtold, and Keim (1998) tackled a problem that is closely related to that of the present article: they were concerned with clustering variables so that similar variables are clustered together in one-dimensional, two-dimensional, and circular display formats. However—unlike the present article—they were not concerned with placing interesting displays in prominent positions.

Section 2 describes a method for ordering variables in scatterplot matrices, so that interesting panels are clustered along the diagonal. I suggest various merit scores and give examples to show that the method yields improved visualizations. Section 3 describes a method for ordering variables in parallel coordinate displays so that interesting panels are visible. Again, I give an example and suggest various merit scores appropriate for parallel coordinate displays. Section 4 follows with some concluding remarks. The Appendix gives details of a suite of R functions implementing the graphical methods and algorithms described here.

2. SCATTERPLOT MATRICES

According to Hills (1969) the first and sometimes only impression gained from looking at a large correlation matrix is its largeness! The same accusation could be leveled at scatterplot matrices. A scatterplot matrix of $p = 10$ variables has 100 panels, each containing a scatterplot of two variables. On today's high resolution screens each panel will be big enough to display a scatterplot of at least a few hundred cases; when the number of cases is substantially larger than this, it might be better to replace the scatterplot with a display based on binning or density estimation. We assume here that the available panel area is large enough to provide an adequate visualization.

An essential feature of scatterplot matrices is that the variable ordering is the same for rows and columns, specifically, the variable appearing on the y-axis in panels from the ith row appears on the x-axis in the ith column. This format is far more informative than a haphazard arrangement of all pairwise scatterplots. One can immediately locate all plots where a particular variable is shown on the horizontal or vertical axis. Because scaling for each axis is kept constant along a row or down a column, one can relate patterns from plot to plot, across rows or down columns. Cleveland (1995) referred to this as "visual linking."

Typically, variables in a scatterplot matrix are arranged in their standard ordering which is simply their ordering in the dataset. Our goal is to permute the variables so that the resulting arrangement of panels is most effective.

2.1 ROBINSON MATRICES

Which of the $p!$ variable permutations yields the most effective scatterplot matrix visualization of the data? Bertin (1983) advocated the use of diagonalization to simplify diagrams. For scatterplot matrices this suggests that we permute the variables so that the most interesting scatterplots appear close to the diagonal, while less interesting scatterplots are far away from the diagonal. Absolute correlation or rank correlation is a reasonable merit measure when we seek to place plots of highly related variables close to the diagonal of the scatterplot matrix. The scatterplot matrix with the permuted variables should be easier to interpret because panels of similar variables appear together in a block.

Suppose m_{ij} is our (symmetric) merit score on the scatterplot of the ith and jth variables. Then we seek a permutation of the variables so that m is nondecreasing as one moves from left to right across a row (and down a column) towards the diagonal of the scatterplot matrix, and nonincreasing as one moves further along the row (and down the column) away from the diagonal. Formally, the matrix $\mathbf{M}$ of permuted m-values has the property that $m_{ij} \leq m_{ik}$ and $m_{ij} \leq m_{kj}$ for $i < k < j$. The matrix $\mathbf{M}$ is then said to have Robinson form (Robinson 1951).

There is a vast literature on this so-called object seriation problem; see, for example, Kendall (1971) and Hubert (1974). Generally, it is only possible to find a permutation that achieves approximate Robinson form. The permutation giving the "closest" approximation to Robinson form could be found using a brute-force permutation search but this becomes computationally infeasible as the number of variables increases. [For example, enumerating all permutations of nine items via the R function `permutations` (package `gregmisc`) takes more than three minutes of system time on a Macintosh G4 at 733 MHz.] Hubert, Arabie, and Meulman (2001, pp. 54–62) described more efficient algorithms based on dynamical programming and suggested various measures of closeness to Robinson form. There are also fast, ad-hoc methods for finding approximate Robinson forms, based on minimal spanning trees and hierarchical clustering. These algorithms will uncover exact Robinson form when it is present (Hubert 1974). However, it is not known how they perform, in general, when Robinson form is not present. In our examples, we use an algorithm due to Gruvaeus and Wainer (1972) based on single-link clustering. Although the permutation of variables yielded does not have any optimal properties, the scatterplot matrix of the permuted variables is frequently dramatically more interpretable than the scatterplot matrix using the standard variable ordering.

2.2 DIAGONALIZING SCATTERPLOT MATRICES

Cluster analysis is more commonly applied to cases in a dataset, here we are clustering variables. Single link cluster analysis is an agglomerative technique. Initially, there are p clusters, one for each variable. As before, let m_{ij} denote the merit score for the ith and jth variables. Then the two variables with the highest m-score are merged into a single cluster. Thereafter, the two clusters containing the pair of variables with the highest m-score are merged until all variables are in the one cluster. The Gruvaeus and Wainer variation uses ordered clusters, whereas in the standard algorithm the objects in a cluster are unordered. Whenever two ordered clusters $A = (a_1, a_2, \ldots a_{n_a})$ and $B = (b_1, b_2, \ldots b_{n_b})$ are merged, the new cluster is one of (1) $(a_1, \ldots, a_{n_a}, b_1, \ldots, b_{n_b})$; (2) $(a_1, \ldots, a_{n_a}, b_{n_b}, \ldots, b_1)$; (3) $(a_{n_a}, \ldots, a_1, b_1, \ldots, b_{n_b})$; and (4) $(a_{n_a}, \ldots, a_1, b_{n_b}, \ldots, b_{n_1})$, whichever gives a new adjacent pair with the highest m-score. I call this algorithm OSL1.

As example, suppose there are six variables and variable pairs ordered by decreasing m-score are (1,5), (4,6), (3,6), (2,6), (1,4), (1,2), (3,4), (4,5), (2,4), (1,6), (5,6), (1,3), (3,5), (2,5), (2,3). To start with there are six clusters. The algorithm proceeds as follows:

Step	*Action*	*Result*
1	Join clusters containing 1 and 5	(1,5) (2) (3) (4) (6)
2	Join clusters containing 4 and 6	(1,5) (2) (3) (4,6)
3	Join clusters containing 3 and 6, placing 3 next to 6.	(1,5) (2) (4,6,3)
4	Join clusters containing 2 and 6, placing 2 next to 4.	(1,5) (2,4,6,3)
5	Join clusters containing 1 and 4, placing 1 next to 2	(5,1,2,4,6,3)

Notice how two clusters are merged together. In Step 4, for instance, we merge the clusters (2) and (4,6,3) to form (2,4,6,3), because $m_{2,4} > m_{2,3}$.

The resulting permutation is (5,1,2,4,6,3) (or the reverse). Note that the result is invariant under monotone transformations of m. Furthermore, the algorithm is easy to program in statistical programming languages and is adaptable to any agglomerative clustering technique.

Algorithm OSL1 is far faster than a permutation search. Sibson (1973) gave an $O(p^2)$ implementation of single-link clustering, therefore OSL1 requires just $O(p^2)$ operations. In fact, computing $\mathbf{M}$ is a far bigger computational burden, requiring $O(p^2 n)$ operations when correlation is used as the m-score.

2.3 Correlation

Correlation-based merit scores are an obvious choice when the goal is to place plots of highly related variables close to the diagonal of the scatterplot matrix. An example is presented in the following that uses absolute rank correlation. The effect of this is to place plots exhibiting a high degree of monotonicity close to the diagonal. Alternatively, one could use Kendall's τ, another rank-based measure of association. The ACE algorithm of Breiman and Friedman (1985) provides another measure of bivariate association, by finding smooth functions of the variables which maximize the (Pearson) correlation.

Breiman and Friedman (1985) described a dataset consisting of measurements on 330 days in 1976 on daily ozone concentration (Ozone) and eight meteorological quantities. Figure 1 shows a scatterplot matrix of the data, with the variables in standard order. With nine variables, the small amount of space allocated to each panel is sufficient to see linearity, curvature, clusters, and outliers among the pairwise variable plots. However, the scatterplot matrix as a whole is almost overwhelming, especially for novice data analysts. We demonstrate that reordering the variables can reduce the apparent complexity of the display.

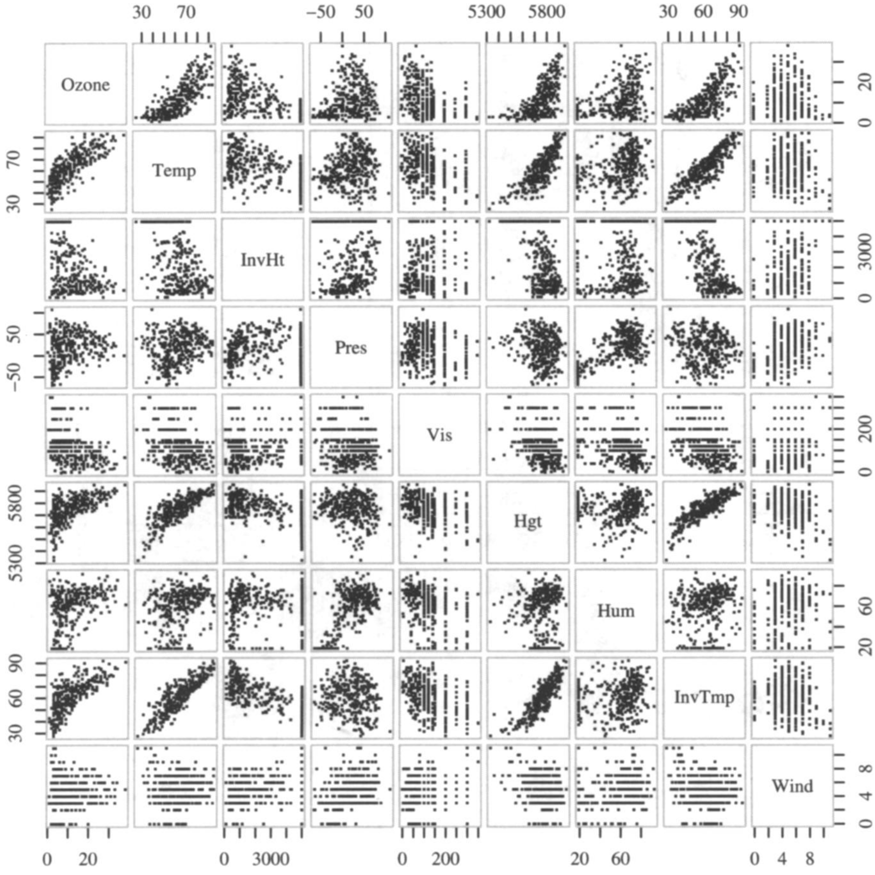

Figure 1. Scatterplot matrix of ozone data. Variables are in standard order.

We score each pair of variables using their absolute rank (Spearman) correlations. With this choice of merit score, **M** is a similarity matrix. We then use algorithm OSL1 to produce a permutation of the variables. Ideally, the matrix **M** of permuted variables has Robinson form.

The scatterplot matrix of permuted variables appears in Figure 2. Panels are divided into three levels of similarity, those with the highest m-score use a gray background, while those with medium and low scores use light gray and white backgrounds, respectively. If the seriation procedure is successful, colors along each row (and down each column) should darken as one approaches the diagonal and lighten as one moves away from the diagonal. In this example there are no serious violations of Robinson form.

The seriation algorithm produces two groups of variables with high pairwise correlations, outlined with a heavy black line in Figure 2. The first group in the upper left-hand corner consists of Hgt, InvTmp, Temp, Ozone, and InvHt and a second group in the lower right consisting of Hum and Press. Of the other two variables, Wind in particular exhibits little similarity to any of the other variables. These conclusions are not as obvious from the standard ordering of variables, which is shown in Figure 1.

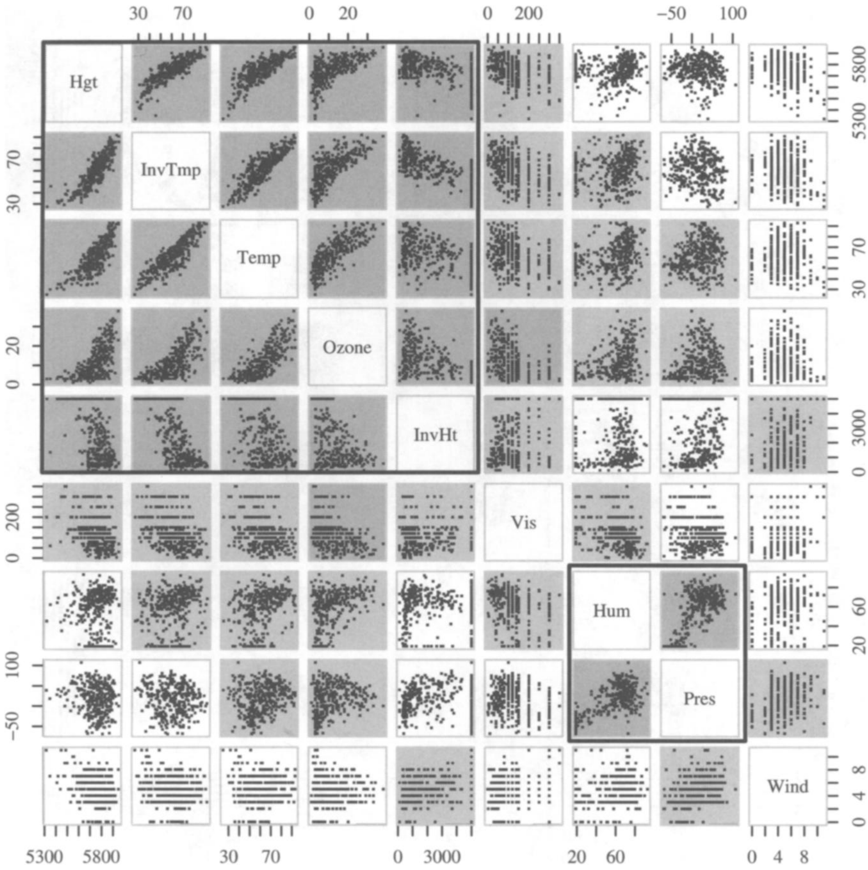

Figure 2. Scatterplot matrix of ozone data. Variables are reordered using OSL1 on matrix of absolute rank correlations. Panel color shows level of absolute rank correlation: gray = top third, light gray = middle third, and white = bottom third.

Reordering the variables so that panels with similar features are clustered together helps us to visually link features across scatterplots. For example, the cluster of points in the bottom left corner of the (InvTmp, Hgt) panel also fall in the bottom left corner of the next two panels in the first row. On those days, variables Hgt, InvTmp, Temp, and Ozone all had low values, but the values for other variables were higher and more spread out.

In the examples throughout this article I use panel color to show the level of the merit index. The colors chosen for the panels should be suitable for use as a background to a scatterplot or a parallel coordinate plot, which may again use color to show groups of cases. Too many color levels may distract from the visual impact of the display. In most examples here we use just three light colors. In the color version of this article (available at www.ingenta.com), I represent the top third of merit values by pink, the middle third by blue, and the bottom third by a very pale yellow. In the gray-scale version, I use a medium gray level for the top third of merit values, a light gray for the middle third, and white for bottom third. The color becomes more pronounced as the merit level increases, so that attention is focused on the panels deemed to be most interesting.

2.4 CLUSTERING CRITERIA

Many simple merit measures relate to the amount of clustering present in a pairwise variable plot. Suppose we consider plots where points are cohesive to be the most interesting. Then we would like m_{ij} to measure the homogeneity in the plot of variables i and j.

Let $d_{ij}(a, b)$ be the Euclidean distance between cases a and b in the plot of variables i and j, assuming the variables are standardized, to unit variance say. Then, we could measure the overall distance between points in plot (i, j) by the total interpoint distance, $d_{ij}^{\text{tot}} = \sum_{a,b} d_{ij}(a, b)$. The quantity d_{ij}^{tot} is a measure of heterogeneity so we take our merit measure m_{ij} to be $-d_{ij}^{\text{tot}}$. This measures how much the points in a plot stick together.

In other situations, the cases divide into g known groups and the most interesting plots are those where the points in a group cluster together and where the groups are distinguishable from each other. Let $d_{ij}(a, b)$ be as before, let G_k denote the kth group and n_k its size. Then, we could measure the overall within-group distance between points in plot (i, j) by $d_{ij}^{gtot} = \sum_k \sum_{a,b \in G_k} d_{ij}(a, b)$. However, this measure gives disproportionate weight to larger groups since the number of terms contributed by each observation depends on its group size. Therefore, it seems more appropriate to use $d_{ij}^{gave} = \sum_k \sum_{a,b \in G_k} d_{ij}(a, b)/n_k$.

Flury and Riedwyl (1988) described a dataset consisting of measurements taken on 100 genuine and 100 counterfeit Swiss bank notes. The measurements are the bottom and top margin widths (Bottom and Top), the left and right edge width (Left and Right), the length of the image diagonal (Diagonal) and the note length (Length). Which of the 15 scatterplots discriminate best between the genuine and counterfeit notes? We award each of the scatterplots the merit score obtained from the group interpoint distance averages, d_{ij}^{gave}. Then the goal is to rearrange the scatterplots so that the group separation increases the closer the plot is to the diagonal.

Figure 3 shows the results when the variables are rearranged using OSL1. The dark gray and black symbols represent the genuine and counterfeit notes, respectively. As before, the panel color codes the level of merit score. The plot of Bottom and Diagonal discriminates best between the genuine and counterfeit notes; in fact, all plots showing the variable Diagonal exhibit a high degree of group separation. The plot of Top and Bottom also exhibits good separation between the groups. All other plots show a moderate to large amount of group overlap. The fake notes are about the right overall length, but the image size is much too small, and all four margins, particularly the bottom margin are oversized by comparison with the genuine notes. Although these conclusions could be reached from the usual scatterplot matrix of variables in standard order, information is gleaned far more readily from the plot of reordered variables where panel color shows the level of clustering.

There are, of course, many other ways of measuring the amount of clustering present in a pairwise variable plot. Gordon (1999, p. 36) and Hubert et al. (2001, p. 19), for example, gave catalogs of various heterogeneity and separation measures used in cluster analysis, which could be used as merit indices in our context. Kaufmann and Rousseeuw (1990) described an index called the "average silhouette width" which is used to measure the amount of clustering present in a dataset partition. They also introduced two measures of clustering strength, namely the agglomerative and divisive coefficients, which are appropriate when the groups are unknown.

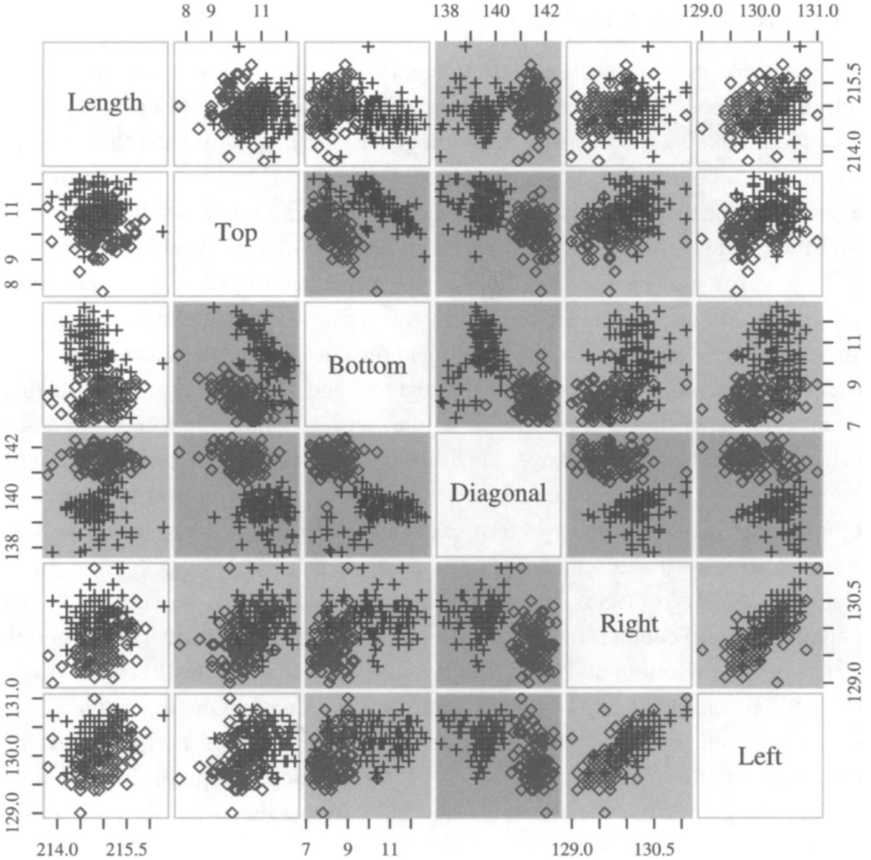

Figure 3. Scatterplot matrix of Swiss bank notes data. Variables are reordered using OSL1 on the matrix of d_{ij}^{gave} scores. Panel color shows merit level: gray = top third, light gray = middle third, and white = bottom third. The dark gray and black symbols represent the genuine and counterfeit notes, respectively.

2.5 Projection Pursuit Indices

Depending on the context, other merit measures may offer more informative visualizations of the data. In the projection pursuit literature, "interestingness" of a projection is measured by its deviation from normality. Various indices were proposed by Friedman and Tukey (1974), Huber (1985), Hall (1989), and Cook, Buja, and Cabrera (1993) among others. In our context, we use a projection pursuit index as the merit measure applied to each pairwise variable plot, and use the seriation algorithm to place interesting plots close to the diagonal of the scatterplot matrix.

Heinz, Peterson, Johnson, and Kerk (2003) described a dataset consisting of 25 body girth and skeletal diameter measurements on 507 individuals. They were interested in exploring the data and developing a regression model relating Weight to the other variables. With 25 variables, there is just about sufficient space on a large computer screen or sheet of paper to display a scatterplot matrix. With 507 points there is a large amount of overplotting, but still one gains a broad impression of bivariate associations and some outliers. However,

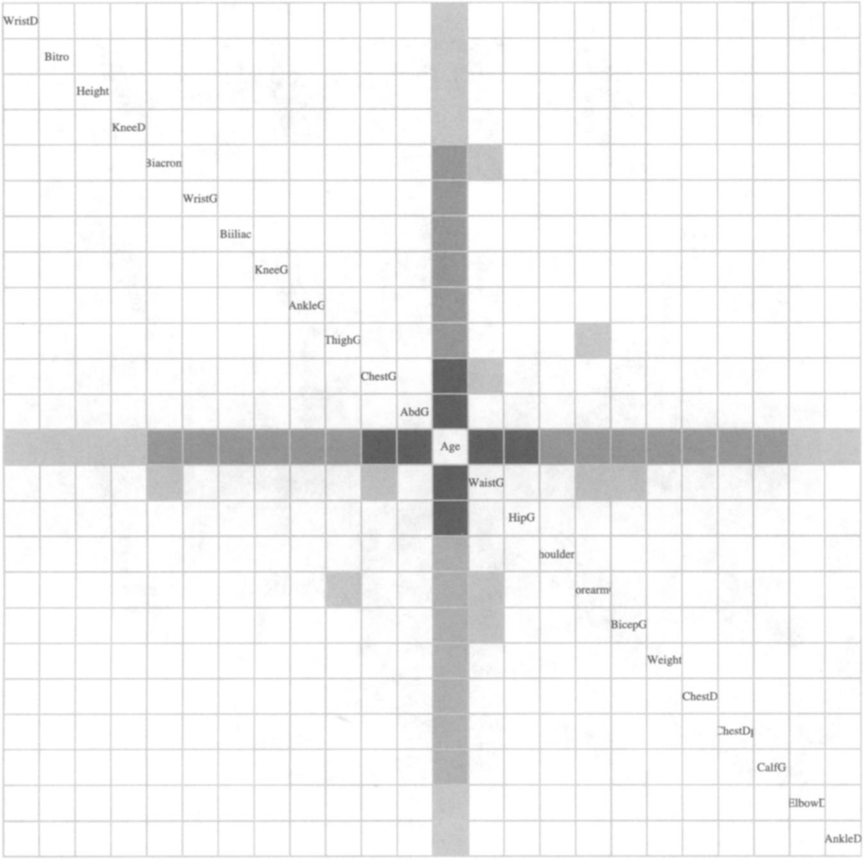

Figure 4. Display of reordered Hermite index values. Index range is cut into six intervals, first three are white, next three are light gray, mid gray, and dark gray, respectively.

there is little point in displaying such a scatterplot matrix in the space available to us here. When space is limited and/or the number of variables is too large, we can simply display the matrix of merit measures rather than the matrix of pairwise variable plots, and then zoom in on interesting sections for a closer examination.

For the body size dataset, we examine the pairwise variable plots for nonnormality. This should highlight any interesting structure apparent in the pairwise variable plots. We use the 2-D Hermite index of Hall (1989) (with four terms) to measure the nonnormality of each pairwise variable plot. Specifically, we use the R/S implementation of this index provided by D. Cook to Statlib (lib.stat.cmu.edu). The index measures the departure of bivariate data from the standard normal distribution, thus the data were sphered prior to computing the index.

Figure 4 is a display of the reordered matrix $\mathbf{M}$ of Hermite index values, with panels colored by cutting the range of merit values into six equal-sized intervals. Panels whose merit value belongs to the lowest three intervals are shown in white, while those whose merit score belongs to the next three intervals are colored light gray, mid gray, and dark

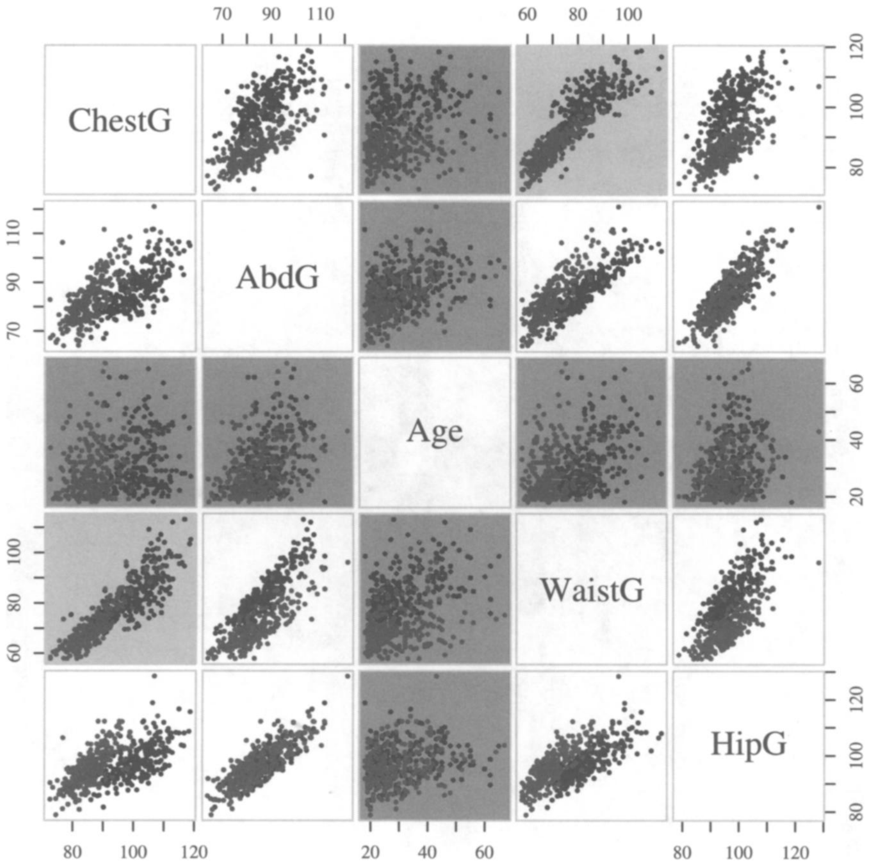

Figure 5. An interesting section of the scatterplot matrix, as identified by the Hermite index. Panels are colored as in Figure 4. Females are in dark gray, males in black.

gray, respectively. The index identifies plots involving Age as the most nonnormal, since all medium and dark gray panels represent Age. Figure 5 shows the middle section of **M** as a scatterplot matrix. Clearly the Age distribution of the participants in the study is skewed, particularly for females. The Hermite index awards a high score to plots with skewness. One might expect that an interestingness index would identify panels where men and women separate into two clusters. Even though such clusters are present in the data, the Hermite index and others in the Cook et al. (1993) package favor panels with skewness.

Obviously, there is no single all-purpose merit measure that highlights panels with "structure" in a scatterplot matrix. For example, a display like that of Figure 4 using correlation as a merit index identifies a block of variables all with high pairwise correlation, but plots with Age are rated as uninteresting. However, if the purpose of the analysis is dimension reduction, or building a regression model, then exploring the correlation structure of the variables is a reasonable starting point. We have suggested some merit measures here, but obviously a good choice of merit index depends on the purpose of the analysis and on the dataset.

3. PARALLEL COORDINATE PLOTS

A parallel coordinates display is a sequence of one-dimensional dot-plots where line segments are drawn to connect the dots pertaining to a particular case. A parallel coordinate display of p variables has $p - 1$ panels, where each panel shows the relationship between a pair of adjacent variables. By comparison with scatterplot matrices parallel coordinate plots are relatively immune to the curse of dimensionality, since the number of panels is $p - 1$ rather than p^2. However, even with moderate numbers of cases the parallel coordinate displays become cluttered and it is hard to see any pattern except for very obvious clusters and outliers. As with scatterplot matrices, variables in a parallel coordinate display are usually arranged in their standard dataset order. Again, we seek to permute the variables so that the resulting parallel coordinate display is most effective.

There are $\binom{p}{2}$ two-variable parallel coordinate displays of p variables, and the p-variable parallel coordinate display contains only $p - 1$ of these. It takes $\lceil (p + 1)/2 \rceil$ parallel coordinate displays to show all pairwise adjacencies (Wegman 1990). For small p we could simply show all of these, but this threatens the data analyst with information overload and becomes impractical for large p. We want to find the permutation of variables giving the most informative selection of panels.

Let m_{ij} be our merit score on the parallel coordinate display of variables i and j. Then, we seek the permutation of variables which maximizes the path length $m_{\text{path}} = \sum_{i=1}^{p-1} m_{i,i+1}$. This is an open-path traveling salesman problem. The standard traveling salesman problem finds the shortest closed path. By inserting an extra node with zero distances to all other nodes, one can use a traveling salesman algorithm to find the shortest open path. The traveling salesman problem is NP-hard which means that a fast (polynomial time) algorithm is unlikely to exist. [See Lawler, Lenstra, Rinnooy Kan, and Shmoys (1985) for a detailed discussion of the traveling salesman problem.] Therefore, even for moderate numbers of variables finding the best permutation is computationally intensive.

3.1 ORDERING VARIABLES

Because our goal is effective data display in a reasonable amount of time, we use a fast algorithm which finds a good permutation of variables rather than an optimal but slow algorithm. There are many heuristic algorithms for finding good, and in some cases near-optimal, traveling salesman routes (see, e.g., Lawler et al. 1985). We use a fast ad-hoc algorithm for finding good permutations based on a minor modification of OSL1.

At each stage, algorithm OSL1 merges the two clusters containing the pair of variables with the highest merit score. When the clusters are merged, this pair of variables may not be adjacent to each other. If the merger occurs early on, the pair of variables will be nearby but not necessarily adjacent in the final ordering, which is reasonable when we are seeking an overall approximate Robinson structure for the scatterplot matrix. For parallel coordinate displays, nearby is not good enough so we merge clusters on the basis of their end (first and last) variables only. Our modified algorithm, which we call OSL2, merges the two clusters containing the pair of end variables with the highest merit score.

For the ordered pairs of Section 2.2, OSL2 proceeds as follows:

Step	*Action*	*Result*
1	Join clusters containing 1 and 5	(1,5) (2) (3) (4) (6)
2	Join clusters containing 4 and 6	(1,5) (2) (3) (4,6)
3	Join clusters containing 3 and 6, placing 3 next to 6.	(1,5) (2) (4,6,3)
4	Join clusters containing 1 and 4, placing 1 next to 4.	(5,1,4,6,3) (2)
5	Join clusters containing 2 and 5, placing 2 next to 5	(2,5,1,4,6,3)

The resulting permutation is (2,5,1,4,6,3) (or the reverse).

We conducted some limited experiments comparing the m_{path} values produced by OSL1 and OSL2 and found that in most cases, OSL2 gives better results than OSL1. To mimic the situations that occur in practice, one should conduct experiments that generate $\mathbf{M}$ matrices obtained from various merit scores applied to data simulated from various multivariate distributions. Rather than conducting such an extensive study, we based our conclusions based on an $\mathbf{M}$ matrix containing values simulated from a Uniform$(0, 1)$ distribution. For 10 variables with 10,000 replications, OSL2 gave the same ordering as OSL1 about 65% of the time, and a better ordering in 26% of cases. The OSL2 value for m_{path} was higher by .23 on average, with a standard deviation of .34. Surprisingly, in about .2% of cases, OSL1 produced orderings with longer paths than the default ordering, but this never happened with OSL2.

With 20 variables, OSL2 performs far better than OSL1. The two algorithms gave the same ordering in only .3% of cases, while OSL2 was better in over 97% of replications. The OSL2 value for m_{path} was higher by 1.12 on average, with a standard deviation of .71.

3.2 Profile Smoothing

Parallel coordinate displays are effective when individual lines are easy to follow across the various panels. This suggests that we seek the permutation of variables that minimizes the number of crossings, or equivalently, maximizes the sum of Kendall's τ correlation between adjacent variables (Griffen 1958). Pearson's correlation could be used instead, especially because it is both easier and faster to compute than Kendall's τ. Alternatively, we could minimize total line lengths, that is, take m_{ij} to be $-d_{ij}^{len}$, where $d_{ij}^{len} = \sum_a |x_{ai} - x_{aj}|$, where $\mathbf{X}$ is the standardized data matrix. Typically, the variables in a parallel coordinate display are prescaled to a common [0,1] interval.

For example, consider the wine recognition database (Blake and Merz 1998). A chemical analysis of 178 Italian wines from three different cultivars yielded 13 measurements.

This dataset is often used to test and compare the performance of various classification algorithms. Figure 6 shows a parallel coordinate display of the data, with variables arranged in standard order. Wines from the three cultivars are distinguished by color, but the lines have a pronounced zig-zag pattern and it is not easy to summarize the difference between the classes.

As with the scatterplot matrix, the $\binom{13}{2}$ different panels are divided into three groups according to their level of merit score. (Note that in a particular parallel coordinate display,

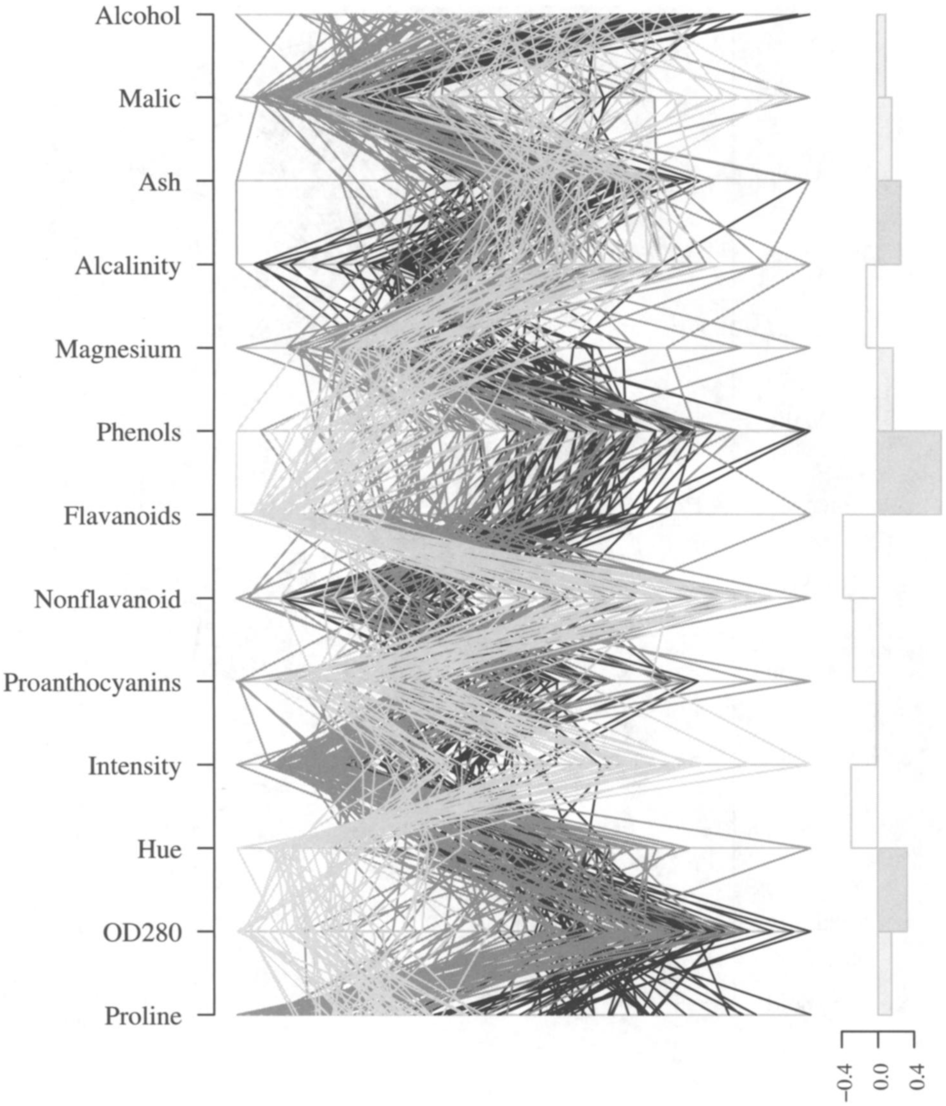

Figure 6. Parallel coordinate plot of Wine data. Variables are in standard order. Line color shows the wine class. Barchart shows the τ value of the adjacent panel. The bars adjacent to panels in the top, middle, and bottom third of τ values (among the 78 possible panels) are colored gray, light gray, and white, respectively.

only 12 of these panels are visible.) Panels whose τ-value are in the top third are assigned a gray color, while those in the middle and bottom thirds are assigned light gray and white. However, using these colors as panel backgrounds would obscure the line colors, so we use a barchart on the right hand side to show the τ value and level of the adjacent panel.

We then rearranged the variables using τ as the merit measure and Figure 7 shows the result. Here the zig-zag pattern is weaker and summarizing the class differences is relatively easy. There is a clear division of the panels into three sections. At the top, (variables Hue

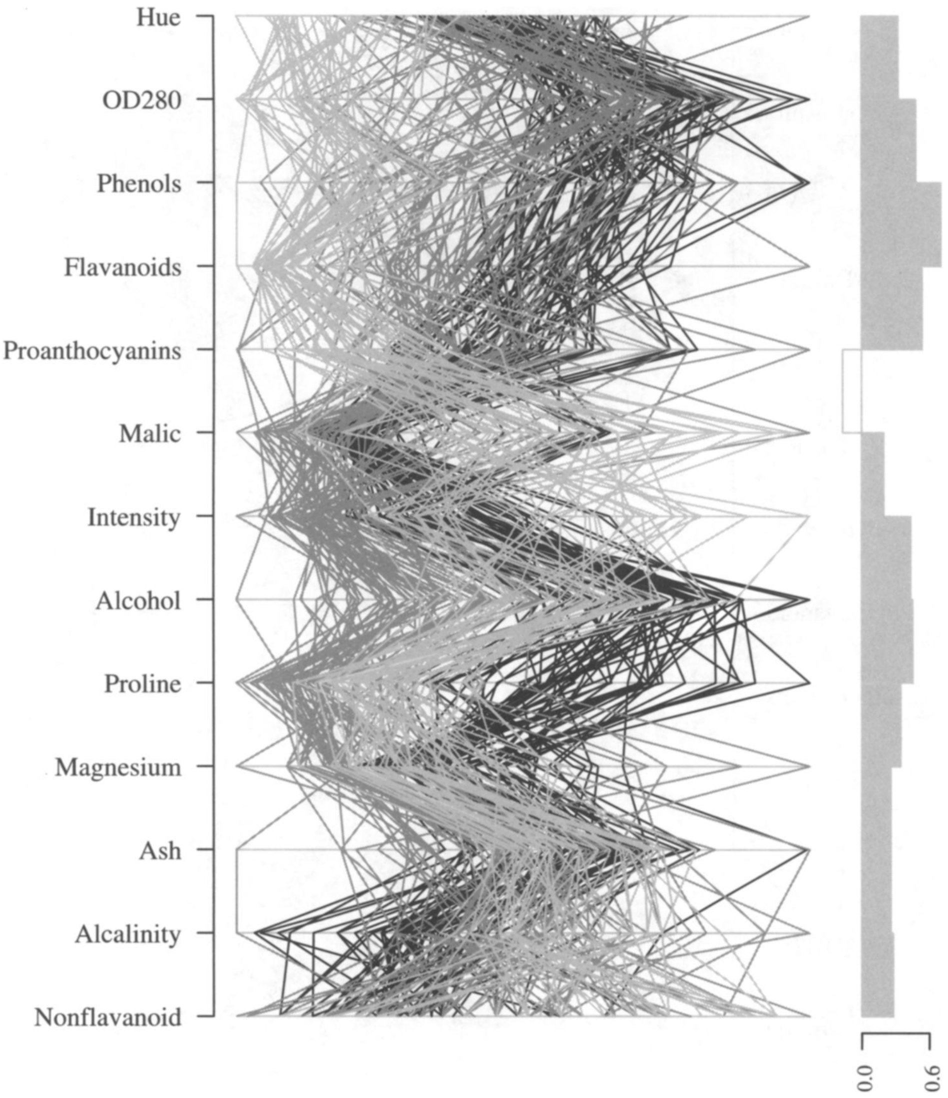

Figure 7. Parallel coordinate plot of Wine data. Variables are ordered by applying OSL2 to matrix of Kendall's τ values. Line color shows the wine class. Barchart shows the τ value of the adjacent panel. The bars adjacent to panels in the top, middle, and bottom third of τ values (among the 78 possible panels) are colored gray, light gray, and white, respectively.

to Proanthocyanins) the left to right class order is light gray (class 3), gray (class 2), and black (class 1). In the middle section (Malic to Magnesium), the gray wines are clearly separated on the left hand side, while lines from the other two groups exchange positions at Alcohol. The classes are scrambled together for the remaining variables. Note that with the standard variable ordering each τ-category is represented among the visible panels, but when the variables are permuted as in Figure 7, only one panel (Proanthocyanins-Malic) has a low value of τ.

In this example, reordering the variables to reduce the number of line crossings made the clusters smoother and more distinguishable from each other. We could also devise a merit measure that measures the "cluster smoothness" in panels of the parallel coordinate display. For this, we take m_{ij} to be $-d_{ij}^{glen}$ where $d_{ij}^{glen} = \sum_k \sum_{a,b,\in G_k} |x_{ai} - x_{bj}|/n_k$, where as before, G_k denotes the kth group and n_k its size. The resulting display (not shown) is similar to Figure 7. The panels divide into three sections containing the same groups of variables as before, except that the order of the variables within the sections is different.

4. DISCUSSION

This article demonstrates that the effectiveness of scatterplot matrices and parallel coordinates displays often improves dramatically when variables are ordered in a systematic way. The overall display becomes more coherent, our ability to visually link different panels improves, and our attention is focused on the more interesting panels of the display.

In the examples presented here, panel color distinguished panels with merit scores in the top third, middle third, and bottom third of values. The R code outlined in the Appendix allows the user to supply any number of colors, or to divide the merit scores by value instead of into equal-sized categories. The ColorBrewer package available from www.colorbrewer.org recommends various color schemes specifically for maps, which may also be appropriate for our context. More generally, the user can simply provide the scatterplot matrix function with a matrix of colors.

This article uses fast, though ad-hoc, ordering algorithms based on single-link clustering. In our experience these give good results, though it might be interesting to conduct experiments comparing these to orderings producing the best Robinson form or the longest path. Alternatively, one could use orderings obtained from other agglomerative clustering algorithms, minimal spanning trees, or principal components.

I suggest various merit measures scoring the component panels in the display. The merit measures fell into one of three categories: (1) correlation-based, (2) measures of cluster cohesion, and (3) nonnormality measures. These indices are suitable for continuous variables, and would not be appropriate for categorical variables with just a few levels. Of course, it is possible to use different ordering methods and merit measures for different subsets of variables, and then to combine the ordered subsets of variables into a single display.

If ordering methods are to be used routinely, the computation should be speedy so that the visualization is produced almost instantaneously. The slowest merit measures proposed in this article required $0(n^2)$ operations for each panel and $0(p^2n^2)$ for the entire display, which could be a considerable overhead for large datasets. It should be possible to develop alternative indices which require fewer operations, perhaps using subsets of cases only.

One could also devise ordering techniques and merit measures for other kinds of data visualizations, such as star glyphs, profile symbol plots, trees, or Chernoff faces (Chambers, Cleveland, Kleiner, and Tukey 1983). In coplots, matrix displays where each panel shows two inner variables for cases partitioned by two outer variables, one could rearrange the levels of the outer (unordered) variables so that interesting panels are in prominent positions, or so that panels with similar patterns are clustered together.

APPENDIX

A suite of R functions implementing the graphical methods and algorithms described here are available from the author and as package `gclus` from http://www.r-project.org. The available functions are divided into three categories: (1) ordering, (2) graphics, and (3) merit measures.

Ordering:

`order.single(merit)`

Implements OSL1. Given scores in a "dist" or matrix, returns an approximate Robinson ordering, used for scatterplot matrices.

`order.endlink(merit)`

Implements OSL1. Given scores in a "dist" or matrix, returns an improved ordering, used for parallel coordinate displays.

Graphics:

`cpairs(data,order=NULL,panel.colors=NULL,...)`

Draws a scatterplot matrix of data. `order`, if present, specifies the order of the variables and `panel.colors`, if present should be a matrix of panel colors. (. . .) are graphical parameters.

`cparcoord(data,order=NULL,panel.colors=NULL,horizontal =FALSE,...)`

Draws a parallel coordinate plot of the data. `order`, if present, specifies the order of the variables and `panel.colors`, if present should either be a vector of panel colors, or a matrix whose i, jth element gives the color for the panel showing columns i and j of data. (. . .) are graphical parameters.

`dmat.color(m,colors = default.mat.colors,byrank=TRUE, breaks=length(colors))`

Given the matrix or "dist" m, returns a matrix of colors. M values are `cut` into categories using `breaks` (ranked distances if `byrank` is `true`) and categories are assigned the values in colors.

Merit Measures:

`colpairs(mat,f,diag=0,...)`

Given an $n \times p$ matrix `mat` and a function `f`, returns the $p \times p$ matrix got by applying `f` to all pairs of columns of `mat`. `Diag`, if present specifies the diagonal value of the returned matrix. `(...)` arguments are passed to `f`.

`partition.crit (x,y,groups,gfun= gave,cfun=sum,...)`

Applies the function `gfun` to each group of x and y values and combines the results using the function `cfun`. `(...)` arguments are passed to `gfun`.

`gave(x,y,...)`

Sums the average distance from each object to all other objects. `(...)` arguments are passed to the function `dist`.

[Received May 2002. Revised November 2003.]

REFERENCES

Ankerst, M., Berchtold, S., and Keim D. A. (1998), "Similarity Clustering of Dimensions for an Enhanced Visualization of Multidimensional Data," in *Proceedings: IEEE Symposium on Information Visualization*, pp. 52–60.

Bertin, J. (1983), *Semiology of Graphics: Diagrams, Networks, Maps*, translated by W. J. Berg, Madison, WI: University of Wisconsin Press.

Blake, C. L., and Merz, C. J. (1998), UCI Repository of Machine Learning Databases [on-line], http://www.ics.uci.edu/~mlearn/MLRepository.html. Irvine, CA: University of California, Department of Information and Computer Science.

Breiman, L., and Friedman, J. (1985), "Estimating Optimal Transformations for Multiple Regression and Correlation," *Journal of the American Statistical Association*, 80, 580–598.

Carr, D. B., and Olsen, A. R. (1996), "Simplifying Visual Appearance by Sorting: An Example using 159 AVHRR Classes," *Statistical Computing and Graphics Newsletter*, 7, 10–17.

Chambers, J. M., Cleveland, W. S., Kleiner, B., and Tukey, P. A. (1983), *Graphical Methods for Data Analysis*, Belmont, CA: Wadsworth.

Cleveland, W. S. (1995), *Visualizing Data*, Summit, NJ: Hobart Press.

Cook, D., Buja, A., and Cabrera, J. (1993), "Projection Pursuit Indexes Bases on Orthonormal Function Expansions," *Journal of Computational and Graphical Statistics*, 3, 225–250.

Flury, B., and Riedwyl, H. (1988), *Multivariate Statistics A Practical Approach*, London: Chapman and Hall.

Friedman, J. H., and Tukey, J. W. (1974), "A Projection Pursuit Algorithm for Exploratory Data Analysis," *IEEE Transactions on Computers*, 23, 881–889.

Friendly, M. (1994), "Mosaic Displays for n-way Contingency Tables," *Journal of the American Statistical Association*, 89, 190–200.

——— (2002), "Corrgrams: Exploratory Displays for Correlation Matrices," *The American Statistician*, 56, 316–324.

Friendly, M., and Kwan, E. (2003), "Effect Ordering for Data Displays," *Computational Statistics and Data Analysis*, 43, 509–539.

Gordon, A. D. (1999), *Classification*, London: Chapman & Hall/CRC.

Griffen, H. D. (1958), "Graphic Computation of Tau as a Coefficient of Disarray," *Journal of the American Statistical Association*, 53, 441–447.

Gruvaeus, G., and Wainer, H. (1972), "Two Additions to Hierarchical Cluster Analysis," *British Journal of Mathematical and Statistical Psychology*, 25, 200–206.

Hall, P. (1989), "Polynomial Projection Pursuit," *The Annals of Statistics*, 17, 589–605.

Heinz, G., Peterson, L. J., Johnson, R. W., and Kerk, C. J. (2003), "Exploring Relationships in Body Dimensions," *Journal of Statistics Education*, 11.

Hills, M. (1969), "On Looking at Large Correlation Matrices," *Biometrika*, 56, 249–253.

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

Hubert, L. (1974), "Some Applications of Graph Theory and Related Non-Metric Techniques to Problems of Approximate Seriation: the Case of Symmetric Proximity Measures," *British Journal of Mathematical and Statistical Psychology*, 27, 134–153.

Hubert, L., Arabie, P., and Meulman, J. (2001), *Combinatorial Data Analysis: Optimization by Dynamic Programming*, Philadelphia, PA: SIAM.

Kaufman, L., and Rousseeuw, P. J. (1990), *Finding Groups in Data: An Introduction to Cluster Analysis*, New York: Wiley.

Kendall, D. G. (1971), "Seriation from Abundance Matrices," in *Mathematics in the Archaeological and Historical Sciences*, eds. F. R. Hodson et al., Edinburgh, Scotland: Edinburgh University Press.

Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G., and Shmoys, D. B. (eds.) (1985), *The Traveling Salesman Problem*, New York: Wiley.

Minnotte, M. C., and West, R. W. (1998), "The Data Image: A Tool for Exploring High Dimensional Data Sets," *ASA Proceedings of the Section on Statistical Graphics*, Alexandria, VA: ASA, pp. 25–33.

Robinson, W. S. (1951), "A Method for Chronologically Ordering Archaeological Deposits," *American Antiquity*, 16, 293–301.

Sibson, R. (1973), "SLINK: An Optimally Efficient Algorithm for the Single Link Cluster Method," *The Computer Journal*, 16, 30–34.

Wegman, E. J. (1990), "Hyperdimensional Data Analysis using Parallel Coordinates," *Journal of the American Statistical Association*, 85, 664–675.