TECHNOMETRICS ©, VOL. 26, NO. 3, AUGUST 1984

The Monotone Smoothing of Scatterplots

Jerome Friedman and Robert Tibshirani

Computational Research Group
Stanford Linear Accelerator Center
and Department of Statistics
Stanford University
Stanford, CA 94305

We consider the problem of summarizing a scatterplot with a smooth, monotone curve. A solution that combines local averaging and isotonic regression is proposed, and we demonstrate its use with two examples. In the second example, the procedure is applied, in a regression setting, to some data from Box and Cox (1964), and it is shown how this new procedure generalizes Box and Cox's well-known family of transformations. In the same example, the bootstrap is applied to obtain a measure of the variability of the procedure.

KEY WORDS: Scatterplot smoothing; Isotonic regression.

1. INTRODUCTION

We consider the following problem: Given a set of n data points $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, how can we summarize the association of the response y on the predictor x by a smooth, monotone function $s(x)$? Put another way, how can we pass a smooth, monotone curve through a scatterplot of y versus x to capture the trend of y as a function of x? This problem is related to both isotonic regression (e.g., see Barlow et al. 1972) and scatterplot smoothing (e.g., see Cleveland 1979).

In this article we propose a solution to the problem that uses ideas from both isotonic regression and scatterplot smoothing (Section 3). This procedure proves to be useful not only as a descriptive tool but also as a method for determining optimal transformations of the response in linear regression (Section 4, Example 2), a method closely related to those of Box and Cox (1964) and Kruskal (1965). We begin with a brief review of isotonic regression and scatterplot smoothing in the next section.

2. A REVIEW OF ISOTONIC REGRESSION AND SCATTERPLOT SMOOTHING

2.1 Isotonic Regression

The problem of isotonic regression on an ordered set is as follows: Given real numbers $\{y_1, y_2, \ldots, y_n\}$, the problem is to find $\{\hat{m}_1, \hat{m}_2, \ldots, \hat{m}_n\}$ to minimize $\sum_1^n (y_i - \hat{m}_i)^2$ subject to the restriction $\hat{m}_1 \leq \hat{m}_2 \leq \cdots \hat{m}_n$. A unique solution to this problem exists and can be obtained from the pool adjacent violators (PAV) algorithm (see Barlow et al. 1972, p. 13). This algorithm is too complex to describe fully here, but the basic idea is the following: Imagine a scatterplot of y_i versus i. Starting with y_1, we move to the right and stop at the first place that $y_i > y_{i+1}$. Since y_{i+1} violates the monotone assumption, we pool y_i and y_{i+1}, replacing them both by their average. Call this average

$$y_i^* = y_{i+1}^* = (y_i + y_{i+1})/2.$$

We then move to the left to make sure that $y_{i-1} \leq y_i^*$—if not, we pool y_{i-1} with y_i^* and y_{i-1}^*, replacing all three with their average. We continue to the left until the monotone requirement is satisfied, then proceed again to the right. This process of pooling the first violator and back-averaging is continued until we reach the right edge. The solutions at each i, $\hat{m}_i$, are than given by the last average assigned to the point at i.

To find the solution for the dual problem ($\hat{m}_1 \geq \hat{m}_2 \cdots \geq \hat{m}_n$), the pool adjacent violators algorithm is applied, starting at y_n and moving to the left. To find $\hat{m}_i$'s to minimize $\sum_1^n (y_i - \hat{m}_i)^2$ subject to non-decreasing or non-increasing $\hat{m}_i$'s, we can choose the best set from the two solutions. We will refer to this two-step algorithm as the pool adjacent violators algorithm.

It is not obvious that the pool adjacent violators algorithm solves the isotonic-regression problem (a proof appears in Barlow et al., p. 12). There are, however, two facts that we can notice about the solution:

- If the data $\{y_1, y_2, \ldots, y_n\}$ are monotone, then $\hat{m}_i = y_i$ for all i; that is, the algorithm reproduces the data.
- Each $\hat{m}_i$ will be an average of y_j's near i. The

average will span over the local nonmonotonicity of the y_i's.

The solution to the isotonic-regression problem is not the solution to the problem of monotone smoothing because the solution sequence $\hat{m}_1, \ldots, \hat{m}_n$ is not necessarily smooth. For example, as we noted, if the data are monotone, the pool adjacent violators algorithm simply reproduces the data; any jaggedness in the data will be passed on to the solution.

In Section 2.2 we briefly review scatterplot smoothing.

2.2 Scatterplot Smoothing

Given n pairs of data points

$$\{(x_1, y_1), \ldots, (x_n, y_n)\}, \qquad x_1 < x_2 < \cdots x_n,$$

assumed to be independent realizations of random variables (X, Y), the goal of a scatterplot smoother is to find a smooth function $s(x)$ that summarizes the dependence of Y on X. (If the x values are not random but fixed by design, we assume that the Y_i's are independent. The derivations are still valid, with expectations over the distribution of X replaced by an appropriate sum.) We assume that Y is some smooth function of X plus a random component,

$$Y = f(X) + \varepsilon, \tag{1}$$

where $E(\varepsilon) = 0$ and $\text{var}(\varepsilon) = \sigma^2 < \infty$. One way to formulate the problem mathematically is to require that $s(x)$ minimize the predictive squared error

$$\text{PSE} = E(Y - s(X))^2, \tag{2}$$

where the expectation is over the joint distribution of (X, Y). If this joint distribution were known, the solution would be

$$\hat{s}(x) = E(Y \mid X = x)$$

for all x. Of course this distribution is rarely known, so the conditional expectation is estimated through local averaging. Many techniques have been suggested for this—the simplest and the one we will make use of is the running mean:

$$\hat{s}_k(x_i) = \text{ave}\,(y_{i-k}, \ldots, y_i, \ldots, y_{i+k}). \tag{3}$$

The windows are shrunken near the left and right endpoints—that is, the set of indexes in a window is actually

$$\{\max(1, i-k), \ldots, i, \ldots, \min(n, i+k)\}.$$

The width of the window over which the average is taken, $2k + 1$, is called the span. Typically, the span is 10%–50% of the observations. To choose the span, a criterion based on the notion of cross-validation can be used. Let $\hat{s}_k^{-i}(x_i)$ denote the running average at x_i, leaving out x_i; that is,

$$\hat{s}_k^{-i}(x_i) = \text{ave}\,(y_{i-k}, \ldots, y_{i-1}, y_{i+1}, \ldots, y_{i+k}), \qquad k \geq 1, \tag{4}$$

with the same endpoint convention as before, Let Z_i be a new observation at x_i—$Z_i = f(x_i) + \varepsilon_i^*$, where ε_i^* is independent of the ε_i's. By using the independence of $\hat{s}_k^{-i}(x_i)$ and y_i, it is easy to show that

$$\frac{1}{n} E \sum_1^n (y_i - \hat{s}_k^{-i}(x_i))^2 = \frac{1}{n} E \sum_1^n (Z_i - \hat{s}_k^{-i}(x_i))^2. \tag{5}$$

Now

$$\hat{s}_k^{-i}(x_i) \approx \hat{s}_k(x_i)$$

and Z_i is independent of $\hat{s}_k(x_i)$, so we have

$$\frac{1}{n} E \sum_1^n (y_i - \hat{s}_k^{-i}(x_i))^2 \approx \frac{1}{n} E \sum_1^n (Z_i - \hat{s}_k(x_i))^2. \tag{6}$$

Since the right side of (6) is an estimate of PSE, a sensible procedure is to choose k to minimize

$$\frac{1}{n} \sum_1^n (y_i - \hat{s}_k^{-i}(x_i))^2.$$

We will denote this value of k by $\hat{k}$.

The resultant smooth $\hat{s}_{\hat{k}}(\cdot)$ thus has the smallest estimated PSE among all estimates of the moving average from (4). Note that

$$\frac{1}{n} E \sum_1^n (Z_i - \hat{s}_k(x_i))^2 = \frac{1}{n} E \sum_1^n (f(x_i) - \hat{s}_k(x_i))^2 + n\sigma^2,$$

so $\hat{k}$ also minimizes an estimate of the expected squared error,

$$\text{ESE}^* = \frac{1}{n} E \sum_1^n (f(x_i) - \hat{s}_k(x_i))^2. \tag{7}$$

Through the use of updating formulas, the optimal span $\hat{k}$ and the smooth $\hat{s}_{\hat{k}}(\cdot)$ can be estimated in a single pass through the data. This speed is important when the algorithm is used as a primitive in other procedures, such as "Projection Pursuit Regression" (Friedman and Stuetzle 1981). For a discussion of running-mean smoothers and more sophisticated smoothers, see Friedman and Stuetzle (1982).

The running-mean smoother produces a smooth function that summarizes the dependence of Y on X, but this function is not necessarily monotone. On the other hand, isotonic regression produces a monotone function that summarizes the dependence of Y on X, but this function is not necessarily smooth. To obtain a smooth, monotone estimate, we can combine the two methods. One way of doing this is proposed in Section 3.

3. MONOTONE SMOOTHING

3.1. The Problem and a Proposed Solution

Suppose we have a set of n data points

$$\{(x_1, y_1), \ldots, (x_n, y_n)\},$$

where $x_1 < x_2 \cdots < x_n$ and our goal is to model, with a monotone function $\hat{m}(\cdot)$, the dependence of y on x. We assume, as in Section 2.2, that the data represent independent realizations of random variables (X, Y) and that Y can be written as $Y = f(X) + \varepsilon$, with $E(\varepsilon) = 0$ and var $(\varepsilon) = \sigma^2 < \infty$. A reasonable property to require of the function $\hat{m}(\cdot)$ is that it should satisfy

$$\begin{aligned} \hat{m}(X) &= \min{}^{-1}\ E_X E_{Z \mid X}(Z_X - \hat{m}(X))^2 \\ &= \min{}^{-1}\ \mathrm{PSE}_M, \end{aligned} \tag{8}$$

subject to $\hat{m}(X)$ non-decreasing in X, where Z_X has the distribution of Z given X. PSE_M is the integrated prediction squared error in predicting the response for a new observation, using the monotone function $\hat{m}(\cdot)$.

As in Section 2.2, we can equivalently minimize the expected squared error

$$\mathrm{ESE}_M = E(f(X) - \hat{m}(X))^2, \tag{9}$$

subject to $\hat{m}(X)$ non-decreasing in X, since $\mathrm{PSE}_M = \mathrm{ESE}_M + n\sigma^2$. It is more convenient to work with ESE_M.

An estimate of ESE_M is

$$\mathrm{ESE}_M^* = \frac{1}{n} \sum_1^n (f(x_i) - \hat{m}(x_i))^2. \tag{10}$$

If we knew $f(\cdot)$, we could simply minimize ESE_M^* over $\hat{m}(\cdot)$ by applying the pool adjacent violators algorithm to $f(\cdot)$. Of course we do not know $f(\cdot)$, but we can estimate it by $\hat{s}_k(x_i)$ (for some k), giving the approximate criterion

$$\mathrm{E\hat{S}E}_M^* = \frac{1}{n} \sum_1^n (\hat{s}_k(x_i) - \hat{m}(x_i))^2. \tag{11}$$

The function $\hat{m}(\cdot)$ minimizing (11) is the output of the pool adjacent violators algorithm applied to $\hat{s}_k(\cdot)$. We will denote this estimate by $\hat{m}_k(\cdot)$. It is the class of smooth, monotone estimates $\hat{m}_k(\cdot)$ that we propose in this article.

There still remains the question of how to choose the span k. A simple way would be to proceed as in Section 2.2, using the $\hat{k}$ that minimizes

$$\sum_1^n (y_i - \hat{s}_k^{-i}(x_i))^2.$$

Another method, suggested by a referee, would be to choose $\hat{k}$ to minimize

$$\mathrm{P\hat{S}E}_M = \sum_1^n (y_i - \hat{m}_k^{-i}(x_i))^2, \tag{12}$$

where $\hat{m}_k^{-i}(\cdot)$ is $\hat{m}_k(\cdot)$ applied to the data set with (x_i, y_i) removed. Using the technique of Section 2.2, one can show that $E(\mathrm{P\hat{S}E}_M) \approx \mathrm{PSE}_M$ so that the resultant $\hat{k}$ minimizes an approximate estimate of (monotone) predictive squared error.

Qualitatively the two techniques will only differ in their choice of span if the underlying function $f(\cdot)$ is far from monotone. In this case, the first method might choose a smaller span to capture curvature in nonmonotone areas of $f(\cdot)$. The second method would choose a larger span because the curvature in the nonmonotone areas is removed by the PAV algorithm.

With this in mind, we compared the two methods in a number of simulated examples. In all cases, there was very little difference in the span chosen. Indeed, it was very difficult to contrive an example in which they differed significantly.

The main difference between the two methods is computational speed. In the first method, the span can be found with a single $O(n)$ pass through the data; once the span is chosen, one application of the pool adjacent violators algorithm is performed. (The PAV algorithm has order between n and n^2, depending on the monotonicity of the sequence.) The second method is far more expensive than the first, requiring n smooths and n applications of the PAV algorithm to obtain the optimal span. In one example with 200 data points, we found that the second method was 100 times slower than the first. The speed of the algorithm is crucial when the procedure is used as a primitive (see Example 2 in Section 4). For this reason, we use the first method for span selection.

3.2 A Summary of the Algorithm

We can summarize the monotone smoothing algorithm as follows:

- *Smooth Y on X*:

$$\hat{s}(x_i) \leftarrow \text{ave}\ (y_{i-\hat{k}}, \ldots, y_i, \ldots, y_{i+\hat{k}})$$

where $\hat{k}$ is chosen to minimize

$$\sum_1^n (y_i - \hat{s}_k^{-i}(x_i))^2.$$

- *Find the closest monotone function to* $\hat{s}_k(\cdot)$: $\hat{m}(\cdot) \leftarrow$ result of the pool adjacent violators algorithm applied to $\hat{s}_{\hat{k}}(\cdot)$.

3.3 Remarks

As a slight refinement of the algorithm, the running-mean smoother was replaced by a running-linear smoother in the examples in Section 4. Running (least squares) lines give results very close to running means in the middle of the data and eliminate some of the bias near the endpoints.

Notice that if the smooth $\hat{s}(\cdot)$ is monotone, then $\hat{m}(\cdot) = \hat{s}(\cdot)$. This makes sense—it just says that the best estimate of $E(Y \mid X)$ in the class of monotone functions is the best estimate over all functions, if the latter is monotone.

In Section 4 we give two examples of the use of this procedure.

4. EXAMPLES

4.1 Example 1

Two hundred points were generated from $y = e^x +$ error, where X was uniformly distributed on [0, 2] and the errors had a normal distribution with mean 0 and variance 1. The result of applying the monotone smoother is shown in Figure 1. A span of 87 points was chosen by the procedure. For comparison, the isotonic regression sequence is also plotted. In this case, the monotone smooth differed only slightly from the smooth (not shown), since the smooth was almost monotone.

4.2 Example 2

In this example we use the monotone-smoothing procedure to find an optimal transformation for the response in a regression. This procedure, similar to that proposed by Kruskal (1965), is a nonparametric version of the Box–Cox procedure (1964). It is also a special case of the "alternating conditional expectation" (ACE) algorithm of Breiman and Friedman (1982). Given a set of responses and covariates

$$\{(y_1, \mathbf{x}_1), \ldots, (y_n, \mathbf{x}_n)\},$$

the goal is to find a smooth, monotone function $\hat{\theta}(\cdot)$ and estimate $\hat{\mathbf{b}}$ to minimize

$$\sum_1^n (\hat{\theta}(y_i) - \mathbf{x}_i \hat{\mathbf{b}})^2 \tag{13}$$

subject to var $(\hat{\theta}(\mathbf{y})) = 1$, where var denotes the sample variance. The procedure is an alternating one, finding $\hat{\mathbf{b}}$ for fixed $\hat{\theta}(\cdot)$ and vice versa.

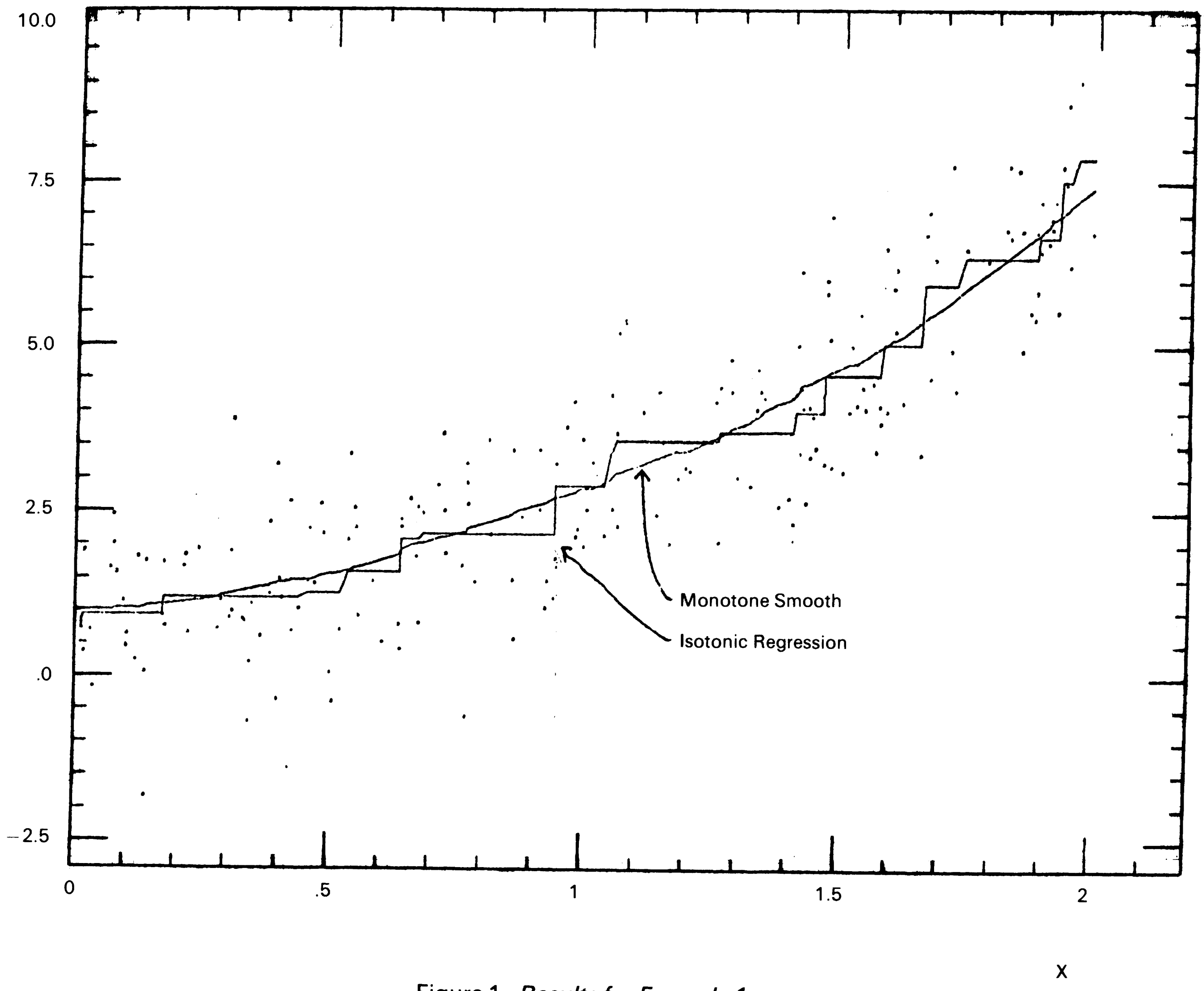

Figure 1. *Results for Example 1.*

```
Initialize   $\hat{\theta}(\cdot) \leftarrow \frac{\mathbf{y}}{\mathrm{var}\,(\mathbf{y})}$.
Repeat
    $\hat{\mathbf{b}} \leftarrow$ least squares estimate of $\hat{\theta}(\cdot)$ on $\mathbf{x}$,
    $\hat{\theta}(\cdot) \leftarrow$ monotone smooth of $\mathbf{x}\hat{\mathbf{b}}$ on $\mathbf{y}$,
    $\hat{\theta}(\cdot) \leftarrow \frac{\hat{\theta}(\cdot)}{\mathrm{var}\,\hat{\theta}(\cdot)}$
until the residual sum of squares (13) fails to decrease.
```

The Kruskal and Box–Cox procedures are essentially variants of the preceding algorithm. Kruskal uses isotonic regression to estimate $\hat{\theta}(\cdot)$, whereas Box and Cox assume that $\hat{\theta}(\cdot)$ belongs to the parametric family

$$g(\lambda) = (y^{\lambda} - 1)/\lambda.$$

We applied this procedure to data on strength of yarns taken from Box and Cox (1964). The data consist of a $3 \times 3 \times 3$ experiment, the response Y being the number of cycles to failure and the factors being the length of test specimen (X_1) (250, 300, or 350 mm), the amplitude of loading cycle (X_2) (8, 9, or 10 mm), and load (X_3) (40, 45, or 50 gm). As in Box and Cox, we treated the factors as quantitative and allowed only a linear term for each. Box and Cox found that a logarithmic transformation was appropriate, with their procedure producing a value of $-.06$ for $\hat{\lambda}$ with an estimated 95% confidence interval of $(-.18, .06)$.

Figure 2 shows the transformation selected by the preceding algorithm. The procedure chose a span of nine observations. For comparison, the log function is plotted (normalized) on the same figure, along with confidence bars corresponding to $g(-.18)$ and $g(.06)$. The similarity is truly remarkable! Figure 3 shows the result of Kruskal's procedure plotted along with the log function. The monotone smooth gives very persuasive evidence for a log transformation, whereas Kruskal's transformation is hampered by its lack of smoothness.

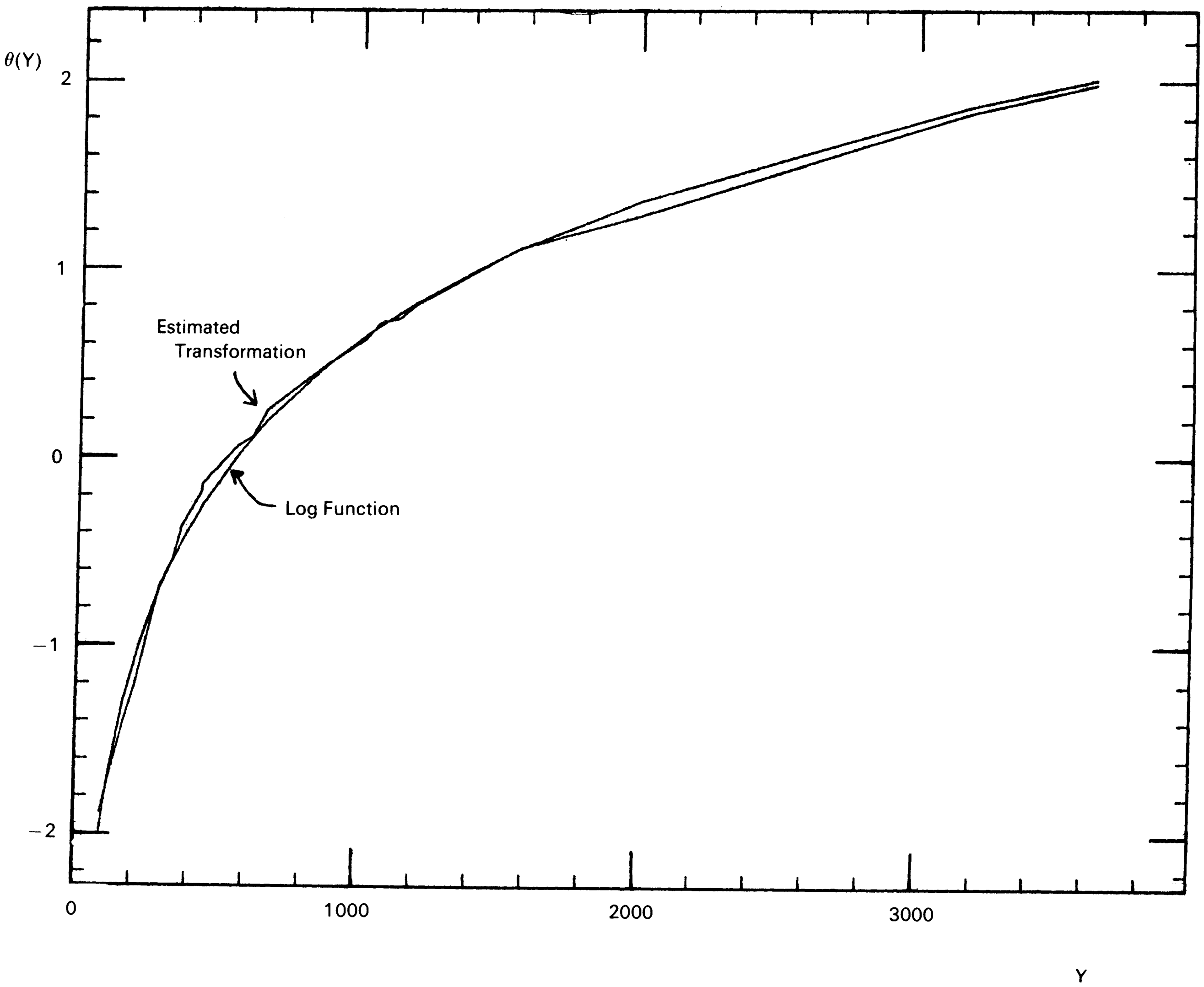

Figure 2. *Monotone Smooth*.

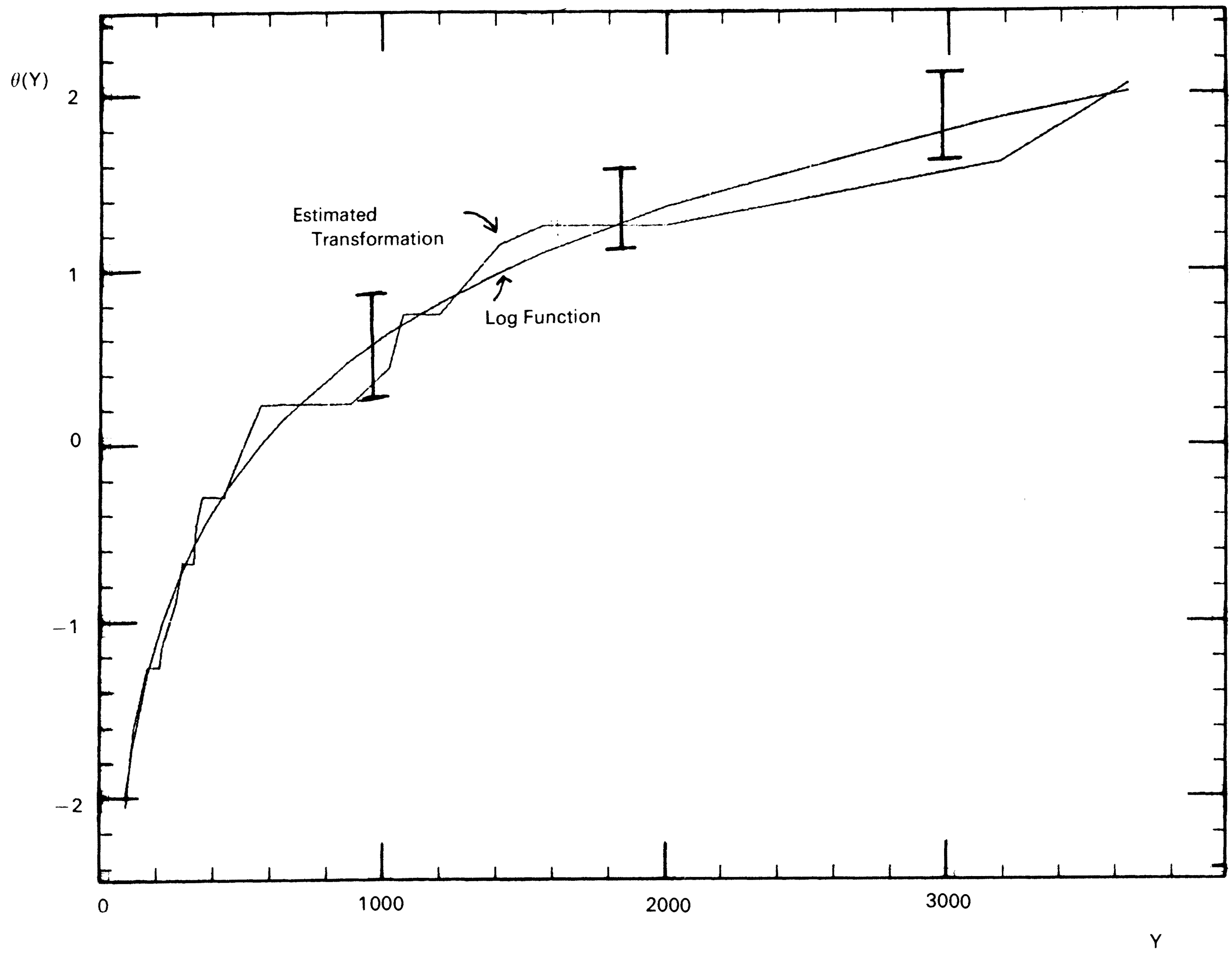

Figure 3. *Kruskal's Algorithm*.

Of course the advantage of the monotone smoothing algorithm is that it does not assume a parametric family for the transformations, and hance it selects a transformation from a much larger class than the Box and Cox family.

To assess the variability of the monotone smooth, we applied the bootstrap of Efron (1979). Since the X matrix in this problem is fixed by design, we resampled from the residuals instead of from the (X, Y) pairs. The bootstrap procedure was the following:

1. Calculate residuals

$$r_i = \hat{\theta}(y_i) - \mathbf{x}_i\hat{\mathbf{b}}, \quad i = 1, 2, \ldots, n.$$

2. Do $j = 1$, NBOOT.
   - Choose a sample $r_1^*, \ldots, r_n^*$ with replacement from $r_1, \ldots, r_n$.
   - Calculate $y_i^* = \hat{\theta}^{-1}(\mathbf{x}_i\hat{\mathbf{b}} + r_i^*), i = 1, 2, \ldots, n.$
   - Compute $\hat{\theta}_j(\cdot)$ = monotone smooth of $y_1^*, \ldots, y_n^*$ on $\mathbf{x}_1, \ldots, \mathbf{x}_n$.
3. End.

NBOOT, the number of bootstrap replications, was 20. It is important to note that, in estimating a common residual distribution via the r_i's, this procedure assumes that the model

$$E(\hat{\theta}(y_i)) = \mathbf{x}_i\hat{\mathbf{b}}$$

is correct (see Efron 1979). The 20 monotone smooths, $\hat{\theta}_1(\cdot), \ldots, \hat{\theta}_{20}(\cdot)$, are shown in Figure 4. The tight clustering of the smooths indicates that the original smooth has low variability. This agrees qualitatively with the tight confidence bars for $g(\lambda)$ in Figure 2.

5. FURTHER REMARKS

The monotone-smoothing procedure that is discussed here should prove to be useful, both as a descriptive tool and as a primitive for any procedure requiring estimation of a smooth, monotone function. It is already being used in the ACE program of Breiman and Friedman (1982).

The use of running-mean or running-linear fits in the algorithm is not essential. Any reasonable smooth

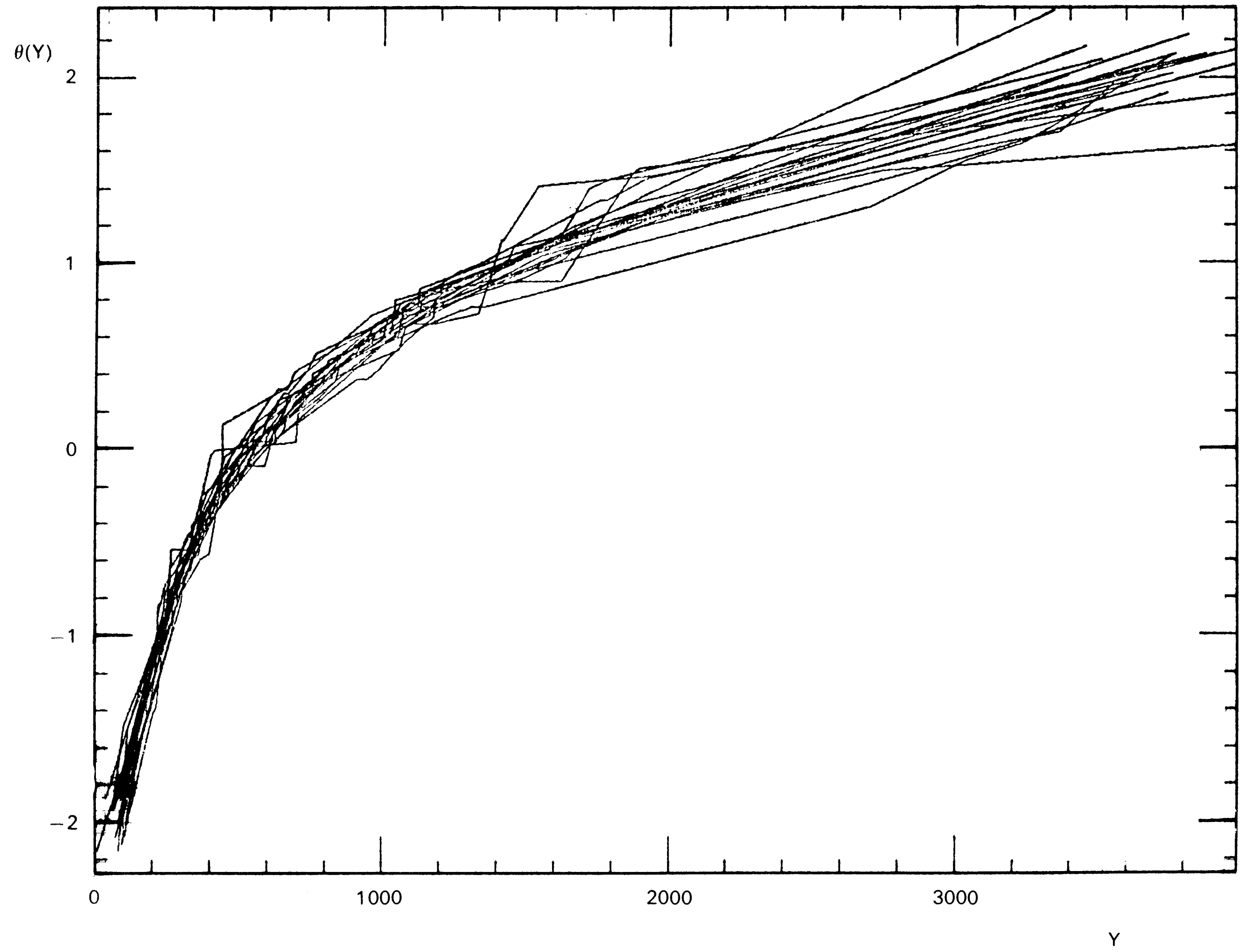

Figure 4. *Bootstrap Smooths*.

(e.g., kernel smoother or cubic splines) should perform equally well.

If robustness to outlying *y* values is a concern, a resistant fit like that proposed in Friedman and Stuetzle (1982) might be used.

The procedure described here is not optimal in any broad sense. It may be possible to develop a one-step procedure that smooths the data using both local information and the global information provided by the monotonicity asumption. Such a procedure might have a slightly lower error of estimation than the monotone smoother described here. But if the procedure is to be used as either a data summary or as a method to suggest a response transformation, we do not think the gain would be worthwhile.

Another way to estimate a monotone smooth would be to apply the pool adjacent violators algorithm first, then smooth the monotone sequence. This has a serious drawback: Though it is true that a running-mean smooth of a monotone sequence is monotone, the running-linear smooth of a monotone sequence is not necessarily monotone. (It is easy to construct a counterexample.) Therefore, one would have to apply the pool adjacent violators again to ensure that the final smooth was monotone. This nonmonotonicity-preserving property is probably true of other popular smoothers. We did not try this procedure, partly because of this fact but mostly because we did not see a sensible justification for it.

We have not presented any arguments (theoretical or otherwise) to suggest that the method proposed here is better than other methods one might use. What is needed for comparison of smoothers is a systematic study over a wide range of situations. A study devoted to smoothers—sort of a sequel to the Princeton location study—would be very valuable in this regard.

ACKNOWLEDGMENTS

The authors thank Trevor Hastie and a referee for their valuable comments. This research was supported in part by the Natural Sciences and Engineering Research Council of Canada, Department of Energy

Contracts DE-AC03-76SF00515 and DE-AT03-81-ER10843, Office of Naval Research Contract ONR N00014-81-K-0340, and U.S. Army Research Office Contract DAAG29-82-K-0056.

[*Received September 1983. Revised February 1984.*]

REFERENCES

BARLOW, R., BARTHOLEMEW, D., BREMNER, J., and BRUNK, H. (1972), *Statistical Inference Under Order Restrictions*, New York: John Wiley.

BOX, G. E. P., and COX, D. R. (1964), "An Analysis of Transformations," *Journal of the Royal Statistical Society*, Ser. B, 26, 211–252.

BREIMAN, L., and FRIEDMAN, J. H. (1982), *Estimating Optimal Correlations for Multiple Regression and Correlation*, technical report Orion 010, Stanford University.

CLEVELAND, W. S. (1979), "Robust Locally Weighted Regression and Smoothing Scatterplots," *Journal of the American Statistical Association*, 74, 828–836.

EFRON, B. (1979), "Bootstrap Methods: Another Look at the Jackknife," *Annals of Statistics*, 7, 1–26.

FRIEDMAN, J. H., and STUETZLE, W. (1981), " Projection Pursuit Regression," *Journal of the American Statistical Association*, 76, 817–823.

——— (1982), *Smoothing of Scatterplots*, technical report Orion 003, Stanford University.

KRUSKAL, J. B. (1965), "Analysis of Factorial Experiments by Estimating Monotone Transformations of the Data," *Journal of the Royal Statistical Society*, Ser. B, 27, 251–263.