SLAC-PUB-3223
STAN-ORION-023
September 1983
M

THE MONOTONE SMOOTHING OF SCATTERPLOTS*

Jerome Friedman and Robert Tibshirani
Stanford Linear Accelerator Center
Stanford University, Stanford, California 94305

ABSTRACT

We consider the problem of summarizing a scatterplot with a *smooth, monotone* curve. A solution that combines local averaging and isotonic regression is proposed. We give some theoretical justification for the procedure and 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 get a measure of the variability of the procedure.

Keywords: *scatterplot smoothing, isotonic regression*

(Submitted to Technometrics)

---

* Work supported by the Department of Energy under contracts DE-AC03-76SF00515 and DE-AT03-81-ER10843, the Office of Naval Research under contract ONR-N00014-81-K-0340, and the Army Research Office under contract DAAG29-82-K-0056.

1. INTRODUCTION

We consider the following problem. Given a set of n data points $\{(x_1, y_1), ...(x_n, y_n)\}$, 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 vs x to capture the trend of y as a function of x? This problem is related to both isotonic regression (see e.g. Barlow et al 1972) and scatterplot smoothing (see e.g. Cleveland 1979).

In this paper 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 $\{y_1, y_2, ...y_n\}$, the problem is to find $\{\hat{m}_1, \hat{m}_2, .. \hat{m}_n\}$ to minimize $\sum_1^n (y_i - \hat{m}_i)^2$ subject to the restriction $\hat{m}_1 \leq \hat{m}_2 \leq ... \hat{m}_n$. A unique solution to this problem exists and can be obtained from the 'pool adjacent violators' algorithm (see Barlow et al, pg. 13). This algorithm is too complex to fully describe here, but the basic idea is the following. Imagine a scatterplot of y_i vs 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 hand edge. The solutions at each i, $\hat{m}_i$, are then given by the last average assigned

to the point at i.

To find the solution for the dual problem ($\hat{m}_1 \geq \hat{m}_2 ... \geq \hat{m}_n$) the pool adjacent violators algorithm is applied, starting at y_n and moving to the left. And to find $\hat{m}_i$'s to minimize $\sum_1^n (y_i - \hat{m}_i)^2$ subject to $\hat{m}_i$'s non-decreasing OR non-increasing, 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's not obvious that the pool adjacent violators algorithm solves the isotonic regression problem— a proof appears in Barlow et al (pg. 12). There are, however, two facts we can notice about the solution:

- if the data $\{y_1, y_2, ...y_n\}$, 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 non-monotonicity 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 $\hat{m}_1 , ... \hat{m}_n$ is not necessarily smooth. For example, as we noted, if the data are monotone, the pool adjacent violators simply reproduces the data; any jaggedness in the data will be passed on to the solution.

In the next subsection, we briefly review scatterplot smoothing.

2.2 Scatterplot Smoothing

Given n pairs of data points $\{(x_1, y_1), ...(x_n, y_n)\}$, $x_1 < x_2 < ...x_n$, assumed to independent realizations of random variables (X, Y), [1] the goal of a scatterplot smoother is find a smooth function $s(x)$ that summarizes the dependence of Y on X. We assume that Y is some smooth function of X plus a random component:

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

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

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

where the expectation is over the joint distribution of (X, Y). If this joint distribution were known, the solution would be $\hat{s}(x) = E(Y|X = x)$ 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:

$$\hat{s}_k(x_i) = Ave(x_{i-k}, ...x_i, ...x_{i+k}) \tag{3}$$

The windows are shrunken near the left and right endpoints— that is, the set of indices in a window is actually $\{max(1, i-k), ...i, ...min(n, i+k)\}$.

The width of the window over which the average is taken, $2k+1$, is called the 'span'. Typically, the span is 10 to 50 percent of the observations. In order to choose the span, a criterion based on the notion of cross-validation can be used. Denote by $\hat{s}_k^{-i}(x_i)$ the running average at x_i leaving out x_i, i.e.

$$\hat{s}_k^{-i}(x_i) = Ave(x_{i-k}, ...x_{i-1}, x_{i+1}, ...x_{i+k}) \tag{4}$$

($k \geq 1$), with the same endpoint convention as before. Let Z_i be a new observation at x_i, i.e. $Z_i = f(x_i) + \epsilon_i^*$ where ϵ_i^* is independent of the ϵ_i's. Then it can be shown that

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

---

[1] If the x values are not random but fixed by design, we would 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.

by using the fact that $\hat{s}_k^{-i}(x_i)$ is independent of y_i. Since the right hand side of (5) 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 $\hat{k}$.

Note also 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 that $\hat{k}$ also minimizes an estimate of the expected squared error

$$ESE^* = \frac{1}{n}E\sum_1^n (f(x_i) - \hat{s}_k(x_i))^2 \qquad (6)$$

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. If we want a *smooth, monotone* function, why not smooth the data first, then apply isotonic regression to the smooth? This is exactly the solution that is proposed in the next section.

3. MONOTONE SMOOTHING

3.1 The Problem and a Proposed Solution

Suppose we have a set of n data points $\{(x_1, y_1), ...(x_n, y_n)\}$, where $x_1 < x_2 ... < x_n$ and our goal is to model, with a monotone function, the dependence of y on x. If we break this problem down into 2 steps

- Find a smooth function $\hat{s}(\cdot)$ that summarizes the dependence of Y on X
- Find the monotone function $\hat{m}(\cdot)$ closest to $\hat{s}(\cdot)$

then using the tools of isotonic regression and scatterplot smoothing discussed in Section 2, the solution is obvious:

- smooth the (X, Y) pairs
- apply the pool adjacent violators algorithm to the smooth

In the next subsection, this heuristic procedure is given some theoretical justification.

3.2 Theoretical Justification for the Procedure

Assume the setup described in Section 2.2. A reasonable property to require of the function $\hat{m}(\cdot)$ is that it should satisfy

$$\hat{m}(X) = min^{-1} E_X E_{Z|X}(Z_X - \hat{m}(X))^2 = min^{-1} PSE_M \tag{7}$$

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 $\hat{m}(\cdot)$. If we knew the true joint distribution of X and Y, or we had a infinite test sample of z_i's, we could minimize PSE_M over $\hat{m}(\cdot)$. Of course, we don't know the joint distribution and we have only a training sample, so we will instead derive a approximate criterion that we can calculate from the training sample alone.

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

$$ESE_M = \frac{1}{n} E \sum_{1}^{n} (f(x_i) - \hat{m}(x_i))^2 \tag{8}$$

since $PSE_M = ESE_M + n\sigma^2$. It turns out to be more convenient to work with ESE_M.

We can first replace the marginal distribution of X by the marginal empirical distribution to obtain

$$ESE_M^* = \frac{1}{n} E \sum_1^n (f(x_i) - \hat{m}(x_i))^2 \tag{9}$$

Clearly, $E_X(ESE_M^*) = ESE_M$, so we can simplify the problem to that of finding an estimate of ESE_M^*.

If we knew $f(\cdot)$, we could simply minimize an estimate of (9), $\frac{1}{n}\sum_1^n (f(x_i) - \hat{m}(x_i))^2$, over $\hat{m}(\cdot)$ by applying the pool adjacent violators algorithm to $f(\cdot)$. Since we don't know $f(\cdot)$, the next best thing is to replace $f(\cdot)$ with our best estimate (in terms of mean squared error) of $f(\cdot)$. In the class of running mean estimates, the best estimate is $\hat{s}_{\hat{k}}(\cdot)$ (from Section 2.2). Hence our approximate criterion is

$$\hat{ESE}_M^* = \frac{1}{n} \sum_1^n (\hat{s}_{\hat{k}}(x_i) - \hat{m}(x_i))^2 \tag{10}$$

To minimize $\hat{ESE}_M^*$ over $\hat{m}(\cdot)$, we simply apply the pool adjacent violators algorithm to $\hat{s}_{\hat{k}}(\cdot)$.

How far off (on the average) will the $\hat{m}(\cdot)$ obtained by minimizing $\hat{ESE}_M^*$ be from the $\hat{m}(\cdot)$ that minimizes ESE_M^*? Unfortunately, it is difficult to get a handle on this. We can expand the expected value of $\hat{ESE}_M^*$ as follows:

$$\begin{aligned}
\frac{1}{n} E \sum_1^n (\hat{s}_{\hat{k}}(x_i) - \hat{m}(x_i))^2 &= \frac{1}{n} E \sum_1^n (\hat{s}_{\hat{k}}(x_i) - f(x_i) + f(x_i) - \hat{m}(x_i))^2 \\
&= \frac{1}{n} E \sum_1^n (\hat{s}_{\hat{k}}(x_i) - f(x_i))^2 + \frac{1}{n} E \sum_1^n (f(x_i) - \hat{m}(x_i))^2 \\
&\quad + \frac{2}{n} E \sum_1^n (\hat{s}_{\hat{k}}(x_i) - f(x_i))(f(x_i) - \hat{m}(x_i))
\end{aligned} \tag{11}$$

Note that only the last two terms in (11) involve $\hat{m}(\cdot)$. If $\hat{s}_{\hat{k}}(\cdot)$ is exactly equal to $f(\cdot)$, then the expected value of $\hat{ESE}_M^*$ is equal to ESE_M^*. Otherwise, we can just hope that since $|\hat{s}_{\hat{k}}(\cdot) - f(\cdot)|$ should be small, the cross product term will be small compared to the 2nd term.

3.3 A Summary of the Algorithm

We can summarize the monotone smoothing algorithm as follows:

- *Smooth Y on X:* $\hat{s}(x_i) \leftarrow Ave(x_{i-\hat{k}}, ...x_i, ...x_{i+\hat{k}})$ 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 pool adjacent violators algorithm applied to $\hat{s}_{\hat{k}}(\cdot)$

3.4 Remarks

- As a slight refinement of the algorithm, the running mean smoother was replaced by a running linear smoother in the numerical examples that follow. 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 good sense— it just says that the best estimate of $E(Y|X)$ in the class of monotone functions is the best estimate over all functions, if the latter is monotone.

In the next section, we give two examples of the use of this procedure.

4. EXAMPLES

Example 1.

200 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.

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 non-parametric 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

$$\sum_{1}^{n} (\hat{\theta}(y_i) - \mathbf{x}_i \hat{\mathbf{b}})^2 \qquad (11)$$

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:

Initialize: $\hat{\theta}(\cdot) \leftarrow \frac{\mathbf{y}}{Var(\mathbf{y})}$

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 $\mathbf{y}$

$\hat{\theta}(\cdot) \leftarrow \frac{\hat{\theta}(\cdot)}{Var\hat{\theta}(\cdot)}$

Until residual sum of squares(11) fails to decrease

Both the Kruskal and Box-Cox procedures are essentially variants of the above algorithm. Kruskal uses isotonic regression to estimate $\hat{\theta}(\cdot)$, while Box and Cox assume that $\hat{\theta}(\cdot)$ belongs to the parametric family $(y^{\lambda} - 1)/\lambda$.

We applied this procedure to data on strength of yarns taken from Box and Cox (1964). The data consists of a 3x3x3 experiment, the response Y being number of cycles to failure, and the factors length of test specimen (X_1) (250, 300 or 350 mm), 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 quantitive 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 percent confidence interval of (-.18,.06).

Figure 2 shows the transformation selected by the above algorithm. The procedure chose a span of 9 observations. For comparison, the log function is plotted (normalized) on the same figure. 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, while Kruskal's transformation is hampered by its lack of smoothness.

The advantage, of course, of the monotone smoothing algorithm is that it doesn't assume a parametric family for the transformations, and hence it selects a transformation from a much larger class than the Box and Cox family.

In order 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:

Calculate residuals $r_i = \hat{\theta}(y_i) - \mathbf{x}_i \hat{\mathbf{b}}$, $i = 1, 2, ...n$
DO j=1, NBOOT
  Choose a sample $r_1^*, ...r_n^*$ with replacement from $r_1, ...r_n$
  Calculate $y_i^* = \hat{\theta}^{-1}(\mathbf{x}_i \hat{\mathbf{b}} + r_i^*)$, $i = 1, 2, ...n$
  - Compute $\hat{\theta}_j(\cdot)$ = monotone smooth of $y_1^*, ...y_n^*$ on $\mathbf{x}_1, ...\mathbf{x}_n$
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 $\hat{\theta}(y) = \mathbf{x}\hat{\mathbf{b}} + r$ is correct (see Efron (1979)). The 20 monotone smooths, $\hat{\theta}_1(\cdot), ... \hat{\theta}_{20}(\cdot)$, along with the original monotone smooth, $\hat{\theta}(\cdot)$, are shown in Figure 4. The tight clustering of the smooths indicate that the original smooth has low variability. This agrees with the short confidence interval for λ given by the Box and Cox procedure.

5. FURTHER REMARKS

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

Some further points:

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

Any reasonable smooth (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 AND the global information provided by the monotonicity assumption. Such a procedure might have slightly lower error of estimation then 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 don't 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: while 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 counter-example). Therefore, one would have to apply the pool adjacent violators again to ensure that the final smooth was monotone. This "non-monotonicity preserving" property is probably true of other popular smoothers. We didn't try this procedure, partly because of this fact but mostly because we didn't see a sensible justification for it.

Acknowlegments

We would like to thank Trevor Hastie for his valuable comments. This research was supported in part by the Natural Sciences and Engineeering Research Council of Canada.

REFERENCES

Barlow, Bartholemew, Bremner and Brunk (1972). Statistical inference under order restrictions. Wiley. New York.

Box, G.E.P., and Cox, D.R. (1964). An analysis of transformations. JRSS B, 26, 211-252.

Breiman, L. and Friedman J.H. (1982). Estimating optimal correlations for multiple regression and correlation. Stanford U. tech. rep. Orion 010.

Cleveland, W.S. (1979). Robust locally weighted regression and smoothing scatterplots. J. Amer. Statist. Assoc., 74 828-836.

Efron, B. (1979). Bootstrap Methods: another look at the Jackknife. Ann. Stat 7, pp 1-26.

Friedman, J.H., and Stuetzle, W. (1982). Smoothing of scatterplots. Stanford Univ. technical report - Orion 003.

Kruskal, J.B. (1965). Analysis of factorial experiments by estimating monotone tranformations of the data. JRSS B, 27, 251-263.

FIGURE 1

Results for Example 1.

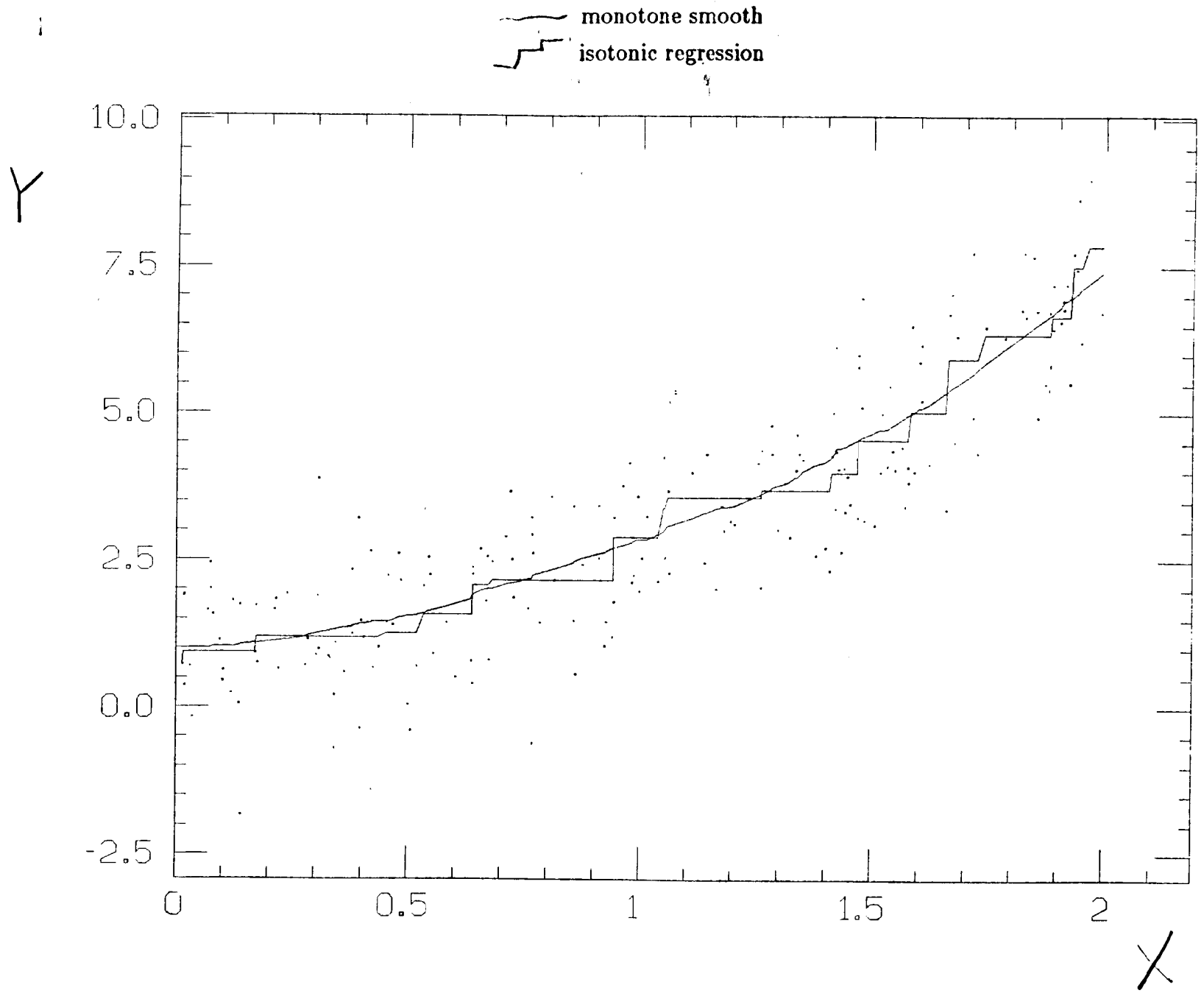

FIGURE 2

Example 2. Monotone Smooth.

estimated tranformation

log function

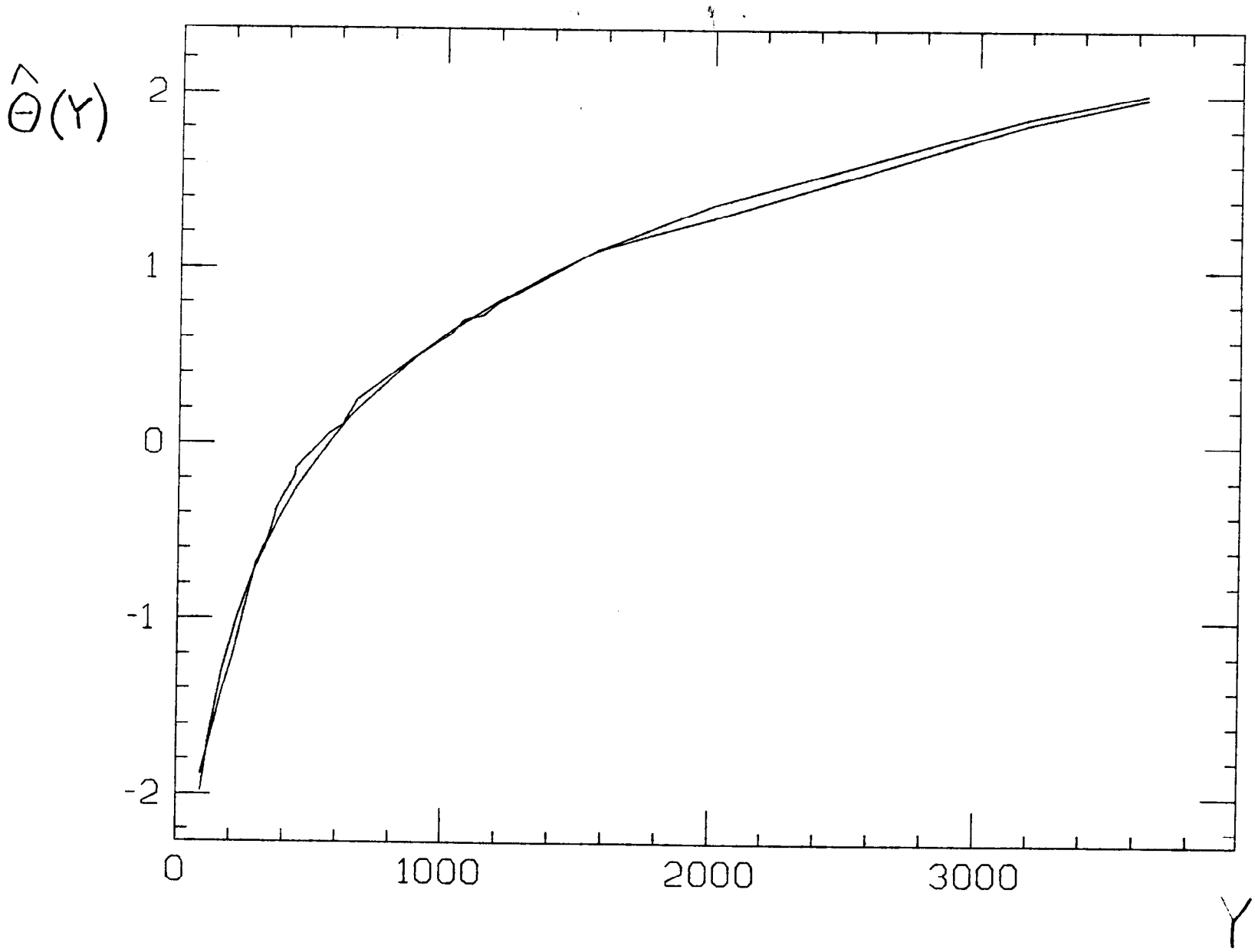

FIGURE 3

Example 3. Kruskal's Algorithm.

estimated tranformation

log function

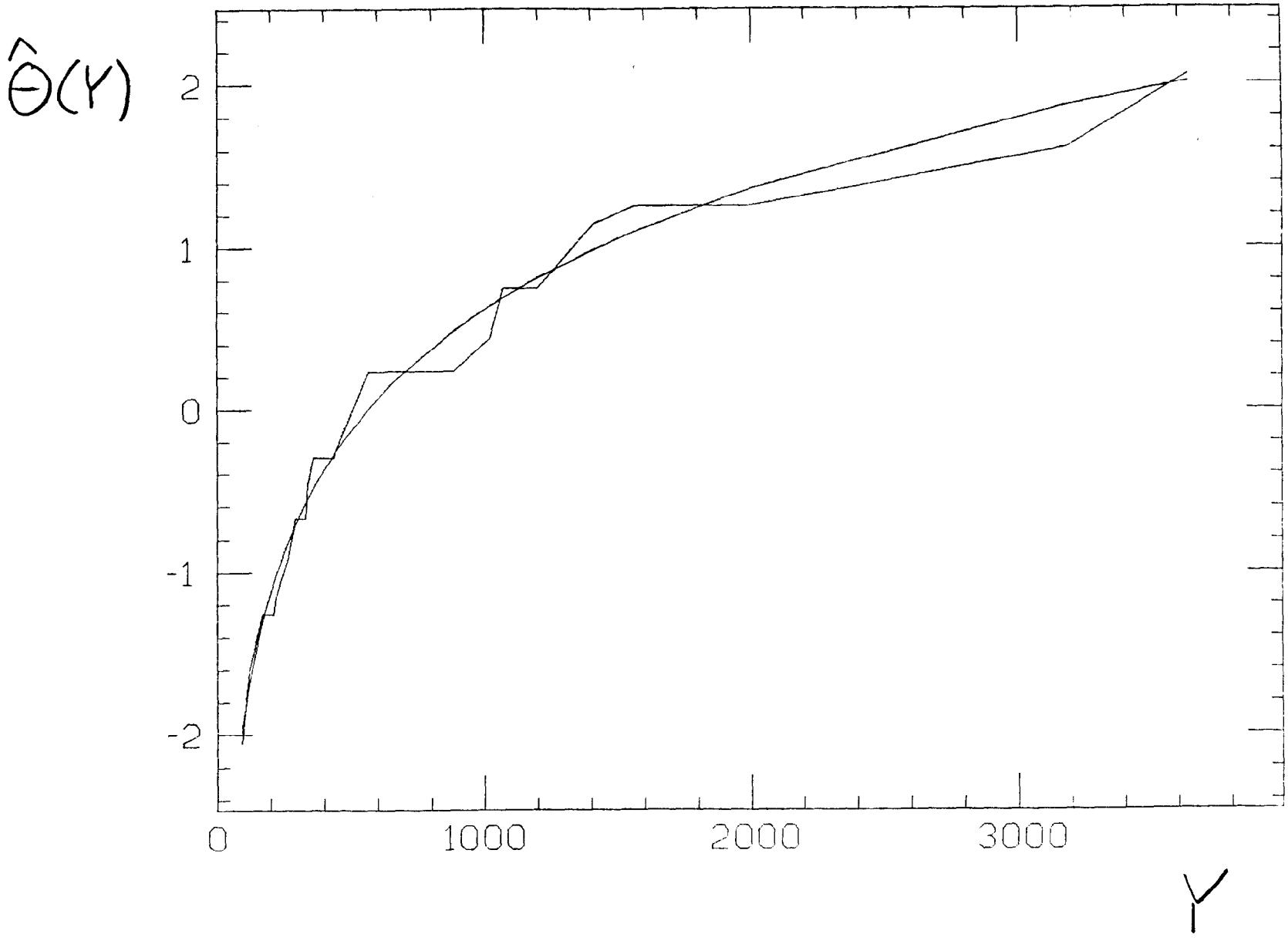

FIGURE 4

Example 2. Bootstrapped smooths.