Statistica Sinica 4(1994), 143-167

PIECEWISE-POLYNOMIAL REGRESSION TREES

Probal Chaudhuri, Min-Ching Huang, Wei-Yin Loh* and Ruji Yao*

*Indian Statistical Institute, National Cheng Kung University and University of Wisconsin-Madison**

Abstract: A nonparametric function estimation method called SUPPORT ("Smoothed and Unsmoothed Piecewise-Polynomial Regression Trees") is described. The estimate is typically made up of several pieces, each piece being obtained by fitting a polynomial regression to the observations in a subregion of the data space. Partitioning is carried out recursively as in a tree-structured method. If the estimate is required to be smooth, the polynomial pieces may be glued together by means of weighted averaging. The smoothed estimate is thus obtained in three steps. In the first step, the regressor space is recursively partitioned until the data in each piece are adequately fitted by a polynomial of a fixed order. Partitioning is guided by analysis of the distributions of residuals and cross-validation estimates of prediction mean square error. In the second step, the data within a neighborhood of each partition are fitted by a polynomial. The final estimate of the regression function is obtained by averaging the polynomial pieces, using smooth weight functions each of which diminishes rapidly to zero outside its associated partition. Estimates of derivatives of the regression function may be obtained by similar averaging of the derivatives of the polynomial pieces. The advantages of the proposed estimate are that it possesses a smooth analytic form, is as many times differentiable as the family of weight functions are, and has a decision tree representation. The asymptotic properties of the smoothed and unsmoothed estimates are explored under appropriate regularity conditions. Examples comparing the accuracy of SUPPORT to other methods are given.

Key words and phrases: Consistency, cross-validation, nonparametric regression, recursive partitioning, smooth partition of unity, tree-structured regression.

1. Introduction

In regression analysis, we typically have n observations on a response variable Y and a vector of K regressors $\boldsymbol{X} = (X_1, \ldots, X_K)$. The response is assumed to depend on the regressors through the relationship $Y = g(\boldsymbol{X}) + \epsilon$, where ϵ is a "noise" component in the model that is assumed to be random with conditional mean zero given any $\boldsymbol{X}$. The aim of a regression analysis is to find an estimate $\hat{g}(\boldsymbol{X})$ of $g(\boldsymbol{X})$ that minimizes a certain loss function (e.g., squared error loss).

Sometimes it may be appropriate to assume that g has a special form, such as

$$g(\boldsymbol{X}) = \beta_0 + \sum_{k=1}^{K} \beta_k h_k(X_k), \tag{1}$$

where the functions $h_1(X_1), \ldots, h_K(X_K)$ are known but the coefficients β_k's are not. This is an example of a parametric regression problem, so called because (1) contains only a fixed number of unknown parameters and the number does not change with the sample size n. Nonparametric regression analysis generalizes the parametric formulation by making weaker assumptions on g. Techniques for obtaining estimates of g, when it is a smooth function of its arguments, include B-splines and smoothing splines (de Boor (1978), Eubank (1988), Ramsay (1988), Wahba (1990)), kernel smoothers (Gasser and Muller (1979, 1984), Nadarya (1964), Rosenblatt (1971), Watson (1964)), and locally weighted regression (Cleveland and Devlin (1988)). Other approaches approximate g with a sum of smooth functions, e.g., projection-pursuit regression (Friedman and Stuetzle (1981)), generalized additive models (Buja, Hastie and Tibshirani (1989), Hastie and Tibshirani (1990), Stone (1985)), and PIMPLE (Breiman (1991)).

In a different direction, attempts have been made to estimate g by recursively partitioning the data and the regressor space, without requiring that the estimate be smooth everywhere. Techniques that yield piecewise constant estimates of g are implemented in the AID (Sonquist (1970)) and the CART (Breiman, Friedman, Olshen and Stone (1984)) regression programs, and those that yield piecewise linear estimates are reported in Breiman and Meisel (1976) and Friedman (1979). These methods have the disadvantage of yielding estimates that are discontinuous. On the other hand, they possess two big advantages that other methods do not have:

1. The decision tree created by recursive partitioning can provide useful information about the regressor variables.
2. The estimated surfaces in each node are simple and functionally explicit (being constants or linear models) and hence are easy to comprehend.

The MARS method of Friedman (1991) produces continuous function estimates by combining spline fitting with recursive partitioning, using the latter to choose the knots. The cost of obtaining continuity is the increased difficulty in interpreting models that are sums of products of splines. Another difficulty is that, because of their complexity, the statistical properties of the MARS estimate are very hard to study analytically.

The main purpose of this paper is to introduce a new method of tree-structured regression called SUPPORT ("Smoothed and Unsmoothed Piecewise-Polynomial Regression Trees") and compare its prediction mean square error

(PMSE) with those of other methods. In addition, we show how the function estimate can be made smooth, thereby overcoming one of the biggest drawbacks of the earlier tree-structured methods. Our technique provides estimates of various derivatives of the unknown regression function as well. The proposed method is built on three principles:

1. Use of polynomial models to fit each subset.

2. Use of a new recursive partitioning algorithm (Huang (1989)) to generate contiguous subsets of the regressor space.

3. Use of weighted averaging to combine the piecewise-polynomial regression estimates into a single smooth one (see O'Sullivan (1991) for a similar suggestion for smoothing CART).

The recursive partitioning algorithm is designed to mimic the likely strategies of a human data analyst. The chief areas in which it differs from CART are as follows.

1. Whereas CART fits a constant to each node, SUPPORT fits linear (or polynomial) regression models. This makes the SUPPORT trees shorter (often substantially shorter) than the CART trees. This is important because the interpretability of a tree diminishes rapidly with the number of levels. For example, consider a data set that is generated by a linear model with one or more nonzero slope coefficients. CART will produce a tree with the number of levels increasing without bounds as the sample size increases. If there are more than two regressors, it is very difficult to infer the underlying simple structure from such trees. The use of linear or polynomial fits also provides another benefit, namely, estimates of the derivatives of the function.

2. Unlike CART, which obtains a regression tree by backwards "pruning" of an overly large tree, SUPPORT uses a cross-validatory multi-step look-ahead stopping rule to determine tree size. This is not necessarily faster, but it is more natural because it resembles what one might do when independent test samples are available. It does not appear to be any less effective than pruning. (See Breiman and Meisel (1976) and Friedman (1979) for earlier versions of one-step look-ahead using random splits.)

3. CART chooses its splits based on the degree of reduction in residual sum of squares (RSS). SUPPORT selects its splits by analysis of the distributions of the residuals. The rationale is that if a fitted model is unsatisfactory, the lack of fit would be reflected in the distributional pattern of the residuals. (See Loh (1991) and Ahn and Loh (1994) for extensions of this idea to regression modeling of censored survival data.)

The regression tree algorithm is motivated and described in the next section. The method of weighted averaging is explained in Section 3. Section 4 gives examples to demonstrate the relative strengths of SUPPORT compared to other methods. The number of regressors in these examples range from two to sixteen. Two very attractive features of our method are that the estimates work quite well for large and complex data sets, and they have simple technical forms (being piecewise-polynomials). The latter allows us to study their large sample behavior theoretically in Section 5. The last section concludes the paper with some remarks.

2. The Recursive Partitioning Algorithm

2.1. Motivation and description

An ideal goal of recursive partitioning regression is to find a regression tree of minimal size and smallest PMSE. The problem is how to measure the true PMSE of a given tree. If a very large independent test set is available, the PMSE of a tree can be accurately estimated from the test sample. In the absence of an independent test set, the next best thing seems to be cross-validation estimation of PMSE using the same data (called the "learning sample") that is used to construct the function estimate.

Because there is usually an error associated with cross-validation, we search for a tree whose estimated PMSE is not reduced by more than a user-specified fraction (f) through splitting of its terminal nodes. In deciding whether a node should be split, it is not enough to determine the change in PMSE after just one split. If the regression function is like a sine-wave with many cycles, for example, it may take several splits of a node before substantial reductions are realized. This means that some form of "looking ahead" is needed. We accomplish this by employing "local" cross-validation at each node. That is, before labeling a node t as terminal, we divide the data in the node into V cross-validation samples. For each combination of $(V-1)$ parts, a nested sequence of trees is constructed and the remaining part is used to determine if any of the trees in the sequence report a fractional reduction of PMSE greater than f. If the proportion of times out of V when the latter event occurs exceeds a pre-assigned threshold level (η), the node t is split.

Split selection is achieved by computing the residuals from a linear model fitted to the node and comparing the distributions of the two subsamples associated with the positive and negative residuals along each regressor coordinate. The motivation for this is that the distributions should not be very different if the fit is adequate. Comparison is made via tests of sample mean and sample variance differences, with the variable giving the most significant test statistic chosen to split the node. The cut-point is the average of the two sample means.

If a quadratic term in X_k is lacking from the model, for example, the difference between the sample variances of the positive and negative residuals along the kth coordinate will be large and would be detected by a test of variance homogeneity (Levene (1960)). This technique was used effectively for classification and survival data modeling in Loh and Vanichsetakul (1988), Loh (1991) and Ahn and Loh (1994).

There are four "tuning parameters" that need to be set in order to execute this part of our algorithm, namely, f, η, V and MINDAT. The latter is the minimum sample size below which a node would not be split. Because sibling subnodes tend to have roughly equal sample sizes, we usually choose MINDAT to be greater than twice the number of regression coefficients fitted in each subnode. We find that the values $V = 10$ and $f = \eta = 0.2$ are generally quite suitable. Given any choice of values for these parameters, the SUPPORT computer program can provide a cross-validation estimate of PMSE for the associated regression tree. Our approach in each of the examples involving real data is to search over a grid of parameter values and choose the tree that possesses the smallest cross-validation estimate of PMSE.

2.2. Algorithmic details

We now present the technical details of the algorithm. Let the ith data vector be denoted by $(y_i, x_{i1}, \ldots, x_{iK})$ and let t_0 denote the root node of the tree.

2.2.1. Split selection

A split of a node t has the form $X_k \leq$ or $X_k > a_k$. Choices of k and a_k are made as follows.

1. *Model fitting.* A linear regression model is fitted to the cases in t. If t contains fewer cases than there are regressors, a constant is fitted.

2. *Residuals.* Each case associated with a nonnegative residual is tagged as "class 1" and each case associated with a negative residual is tagged as "class 2." Let I_j denote the number of class j cases in the node.

3. *Tests for means and variances.* Let $\bar{x}_{\cdot kj}$ and s^2_{kj} denote the mean and the variance of the X_k data values in t that belong to class j, and let $s^2_{k\cdot}$ denote the pooled variance estimate. Defining $z_{ikj} = |x_{ikj} - \bar{x}_{\cdot kj}|$, let $\bar{z}_{\cdot kj}$ and w^2_{kj} denote the mean and the variance, respectively, of the z's. Compute the following two test statistics:

(a) Test for difference in means

$$t^{(1)}_k = (\bar{x}_{\cdot k1} - \bar{x}_{\cdot k2}) \Big/ s_{k\cdot}\sqrt{I_1^{-1} + I_2^{-1}}, \quad k = 1, \ldots, K;$$

(b) Test for difference in variances (Levene (1960))

$$t_k^{(2)} = (\bar{z}_{\cdot k1} - \bar{z}_{\cdot k2})\Big/ w_{k\cdot}\sqrt{I_1^{-1} + I_2^{-1}}, \quad k = 1, \ldots, K.$$

These tests are intended to be sensitive to differences in the two group means and variances, respectively. For example, if the regression function is convex in the node, the group associated with the nonnegative residuals will tend to have larger variance than the group with the negative residuals. Let α'_k be the smaller of the two P-values associated with these two statistics obtained from the Student's t-distribution with $(I_1 + I_2 - 2)$ degrees of freedom.

4. *Variable selection.* Let k_0 be the smallest value of k such that $\alpha'_{k_0} = \min_k \alpha'_k$ and define

$$\alpha(t) = \alpha'_{k_0} \tag{2}$$

to be the smallest P-value for node t. The node is split into two subnodes along $X_{k_0} = a_{k_0}$, where $a_k = (\bar{x}_{\cdot k1} + \bar{x}_{\cdot k2})/2$ is the average of the two class means.

2.2.2. Stopping rule

The decision to split a node or not is made through cross-validation. Let $\mathcal{L}(t)$ denote the learning sample in node t. The cases in $\mathcal{L}(t)$ are randomly divided into V roughly equal parts, $\mathcal{L}_1, \ldots, \mathcal{L}_V$, say (we omit reference to t because it is constant in the present discussion). The following process is repeated for each $v = 1, \ldots, V$:

1. A large regression tree T_v (where the sample size in each terminal node is not greater than MINDAT) is constructed for the cases in $\mathcal{L}^{(v)} = \mathcal{L} - \mathcal{L}_v$. Each intermediate node t' of T_v has a smallest P-value $\alpha(t')$ according to the definition (2). Suppose that there are $s(v)$ such distinct P-values. Adjoin the numbers 0 and 1 to this set and denote them in sorted order by $\{\beta_0, \ldots, \beta_{s(v)+1}\}$, with $\beta_0 = 0$ and $\beta_{s(v)+1} = 1$.

2. Compute $\gamma_i = (\beta_i + \beta_{i+1})/2$, for $i = 1, \ldots, s(v)$.

3. Let $T_{v,i}$ be the subtree obtained by removing the subnodes of all those nodes t' for which $\alpha(t') > \gamma_i$. Because γ_i is increasing in i, this yields a nested sequence of subtrees $T_{v,1} \prec \cdots \prec T_{v,s(v)} = T_v$.

4. Let $\xi(v,i)$ denote the estimate of the PMSE of $T_{v,i}$ obtained by running the test sample $\mathcal{L}_v$ through it. Let f $(0 < f < 1)$ be a user-controlled parameter.

(a) Set $i = 1$.

(b) If $\xi(v,i) < (1-f)\xi(v,1)$, set $\theta(v) = 1$ and exit.

(c) Otherwise increment i by 1.

   i. If $i \le s(v)$, go to Step 4b.

ii. Else, set $\theta(v) = 0$.

The purpose of these steps is to determine if there is a nontrivial tree, obtained by splitting the root node t^* of T_v with a substantially smaller (as defined by f) PMSE than that of the trivial tree consisting of t^* itself.

Once the values of $\{\theta(1), \ldots, \theta(V)\}$ are obtained, define their average $\bar{\theta} = V^{-1} \sum_{v=1}^{V} \theta(v)$. This measures the frequency with which the best cross-validation trees are nontrivial. Let η $(0 < \eta < 1)$ be a pre-selected threshold frequency.

- If $\bar{\theta} > \eta$, the node t is split and the procedure is applied recursively to its children nodes.
- Otherwise, t is declared to be a terminal node.

3. Smoothing by Weighted Averaging

Although the prediction error of the piecewise-polynomial estimate is typically very good (see Huang (1989) and the examples below), it lacks continuity. If the true function is continuous, a continuous estimate would be desirable. One way to produce a continuous estimate is by averaging the polynomial pieces. First note that each polynomial is well-defined and continuous (in fact, infinitely differentiable) not only on the subset of the regressor space over which it is constructed, but over the whole space. Therefore if we use smooth weight functions in the average, where each weight function drops rapidly to zero outside its associated partition, a smooth estimate will result.

Specifically, let $\tau > 0$ be a small number. For each partition t (corresponding to a terminal node of a tree), let t_τ be a set containing t such that the volume of t_τ is equal to $(1+\tau)^K$ times that of t. (Since t is typically a Cartesian product of intervals, a simple way to do this is to let t_τ be the corresponding Cartesian product with each side lengthened proportionately.) Suppose that $\hat{g}_t(\boldsymbol{X})$ is a polynomial fitted to the data in t (or to the data in t_τ if that is preferred). Clearly, $\hat{g}_t(\boldsymbol{X})$ is well-defined over the entire regressor space. Let $H(\boldsymbol{X}, t)$ be a bounded nonnegative smooth function associated with t such that $H(\boldsymbol{X}, t) > 0$ for $\boldsymbol{X} \in t$ and is equal to 0 if $\boldsymbol{X}$ is outside t_τ. Define $W(\boldsymbol{X}, t) = H(\boldsymbol{X}, t) / \sum_s H(\boldsymbol{X}, s)$. Then the weighted average

$$g^*(\boldsymbol{X}) = \sum_t W(\boldsymbol{X}, t) \hat{g}_t(\boldsymbol{X}),$$

which is defined on the entire regressor space in view of the extended definition of $\hat{g}_t(\boldsymbol{X})$, is a smooth estimate that is as many times differentiable as the weight functions are. In the case that $K = 1$ and $t = (a, b)$, for example, one possibility is to take $d = \tau(b-a)$, $t_\tau = (a-d, b+d)$ and set

$$H(x,t) = \left[1 - \{x - (a+b)/2\}^2 \{(b-a)/2 + \delta\}^{-2}\right]^{p+1} I(x \in t_\tau),$$

which is p-times differentiable. An example of an infinitely differentiable weight function is

$$H(x,t) = \exp\Big\{ - d(|x-a+d|^{-1} + |x-b-d|^{-1})\Big\} I(x \in t_\tau). \tag{3}$$

The latter is graphed in Figure 1 for two values of τ.

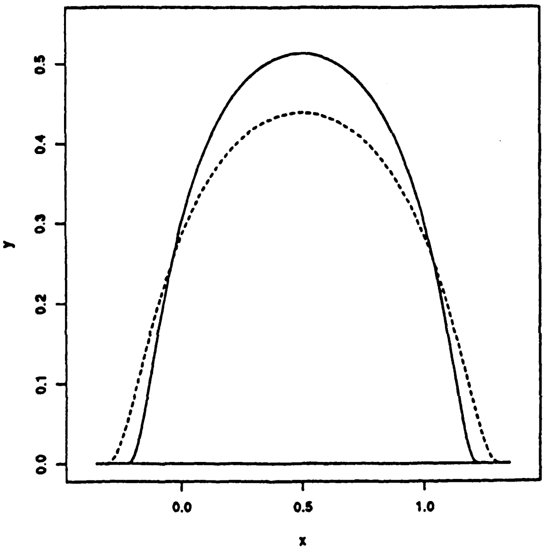

Figure 1. Weight functions (3) with $a = 0$ and $b = 1$. The solid and dashed curves are for $\tau = 0.25$ and 0.35 respectively.

This technique generalizes naturally to higher dimensions by defining the weight functions as products of univariate weights. The estimate tends to be smoother if piecewise-quadratic or cubic fits are used instead of piecewise-linear fits. This is consistent with the well-known nice properties of cubic splines used in data smoothing. The main disadvantages in using high-order polynomials are greater difficulty in interpreting the coefficients and shorter tree structures than sometimes desired.

The same method can be used to estimate the derivatives of the function. Instead of averaging the piecewise-polynomial fits, we average their derivatives. Suppose that the function g is continuously differentiable with derivatives up to order $m \geq 0$ on an open set A that covers the range of the $\boldsymbol{X}_i$'s (the 0-th order derivative of a function is the function itself and any continuous function by definition has a continuous 0-th derivative). We write $\boldsymbol{U}$ to denote the collection of all vectors $\boldsymbol{u} = (u_1, u_2, \ldots, u_K)$ with nonnegative integer coordinates such that $[\boldsymbol{u}] = u_1 + u_2 + \cdots + u_K \leq m$. For $\boldsymbol{u} \in \boldsymbol{U}$, we denote by $D^{\boldsymbol{u}}$ the partial differential operator $\partial^{[\boldsymbol{u}]}/\partial x_1^{u_1} \partial x_2^{u_2} \cdots \partial x_K^{u_K}$ operating on g. If $D^{\boldsymbol{u}} \hat{g}_t(\boldsymbol{X})$ denotes the $\boldsymbol{u}$th derivative of $\hat{g}_t(\boldsymbol{X})$ w.r.t. $\boldsymbol{X}$, the $\boldsymbol{u}$th derivative of $g(\boldsymbol{X})$ may be estimated by

$$\sum_t W(\boldsymbol{X}, t) D^{\boldsymbol{u}} \hat{g}_t(\boldsymbol{X}). \tag{4}$$

Our experience suggests that the values $\tau = 0.25$ or 0.35 are usually quite good. Adaptive selection of τ can be based on cross-validation error estimation.

4. Examples

Three examples are given in this section to illustrate the method. The first uses simulated data with two covariates, so that the results may be presented in 3-D perspective plots. The other two examples involve real data with three and sixteen covariates. For these two examples, we randomly divide the data into two subsets, one to serve as learning sample and the other as test sample. We use the tuning parameter values that minimize the ten-fold cross-validation estimate of PMSE in each case. Although the results are by no means definitive, the examples are indicative of

1. the predictive accuracy of the method,
2. the sensitivity of the tree structures to choice of tuning parameter values,
3. the robustness of the tree structures against perturbations in the data, and
4. the effect of outliers on prediction error.

4.1. Simulated data: 100 learning cases and 2 regressors

The regression function in this example is the bivariate normal density function over the range $X_1, X_2 \in (-1, 1)$. Data were simulated over an equally spaced grid of 100 points according to the formula

$$Y = \exp\{-(X_1^2 + X_2^2)/2\} + 0.2Z$$

where Z is a standard normal variate representing the noise. The true function and the data are plotted in the first row of Figure 2. (There is an inevitable small amount of smoothing in the plots because of the interpolative nature of all 3-D graphing programs). Because of the amount of noise in the data, the concave nature of the true function is hard to detect from the data plot. Using piecewise linear fits with parameter values $f = \eta = 0.2$ and 10-fold cross-validation, our recursive partitioning algorithm partitioned the sample space into eight rectangles. The plots in the second row of Figure 2 show two smoothed estimates obtained

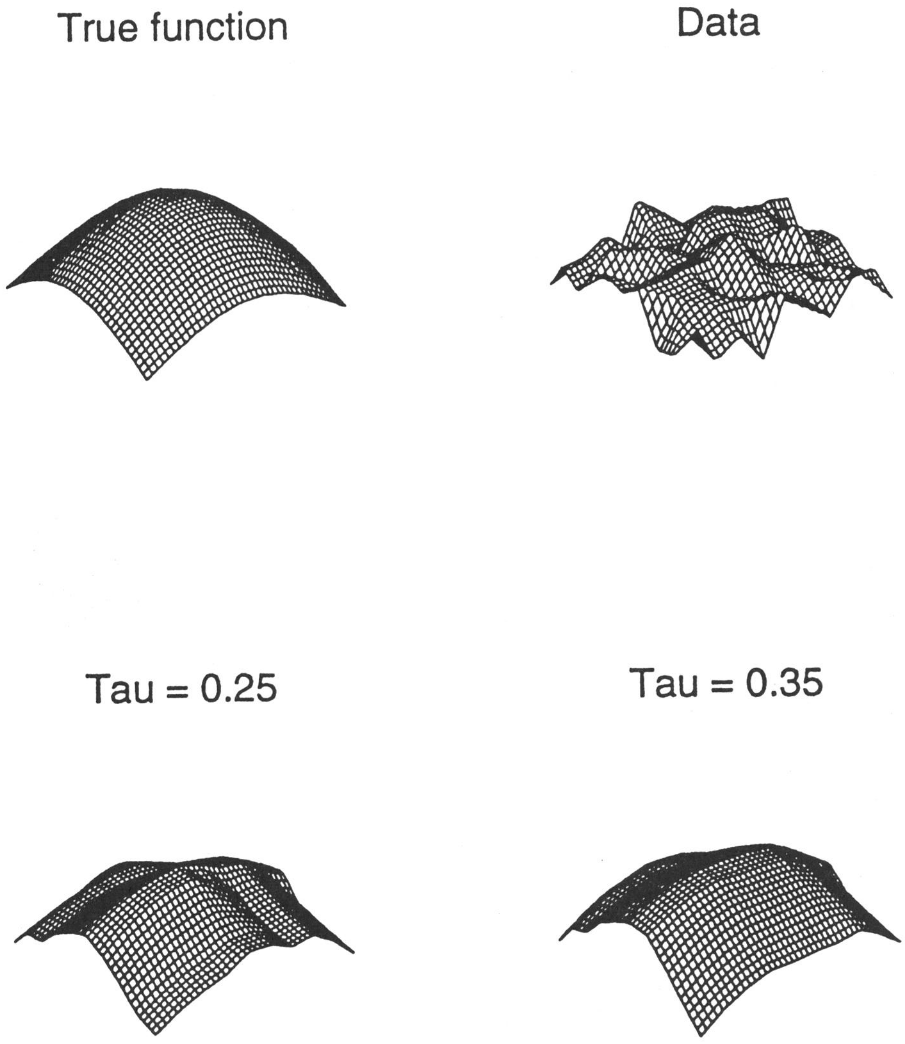

Figure 2. Simulated data example. The upper left figure shows the true regression function and the upper right the data. The two lower figures show the smooth estimates with $\tau = 0.25$ and 0.35. First degree polynomials were used to determine the partitions and piecewise-quadratic fits were averaged to give the final estimates.

by weighted averaging of quadratic polynomials (without cross-product terms) fitted to each partition. The weight function (3) was used. The concavity of the true surface is recovered quite satisfactorily.

4.2. Mumps data: 600 learning cases and 3 regressors

In the following example, we analyze some data on the incidence of mumps in each of the 48 contiguous states of the U.S. (excluding the District of Columbia) from 1953 to 1989. The data on number of mumps cases reported come from the `statlib` archive at Carnegie-Mellon University. There are 1523 observations on four variables. The dependent variable (Y) is the natural logarithm of the number of mumps cases reported per million population in each state. Because we could not get the state populations for every year, we use the 1970 census figures from the `states` database in the S (Becker, Chambers and Wilks (1988)) statistics package. The regressor variables are year (coded as actual year minus 1900) and the longitude and latitude of each state's center (the latter also obtained from S). Longitudes are measured in negative degrees west of the International Date Line. Our goal is to model the spatial and temporal features in the data using piecewise-linear models. The 1523 cases are randomly divided (using the MINITAB software) into a learning sample of 600 cases and a test sample of 923.

Table 1 shows the results of an ordinary least squares (OLS) fit of the learning sample. Year is highly statistically significant, followed by Latitude. The test-sample estimate of PMSE based on the 923 independent observations is 1.956.

Table 1. Coefficients from a linear regression of log-rates of mumps on Year, Latitude and Longitude; $R^2 = 60.8\%$; residual standard deviation = 1.338 with 596 degrees of freedom.

Variable	Coefficient	S.E.	t-ratio
Constant	13.1025	0.6970	18.8
Year	−0.154420	0.0052	−29.7
Latitude	0.071007	0.0123	5.77
Longitude	−0.001640	0.0037	−0.447

Using a MINDAT value of 30, 10-fold cross-validation estimates of PMSE are obtained over a square grid of f and η values from 0.1 through 0.5, with increments of 0.1. The estimates are minimized at $(f, \eta) = (0.1, 0.3)$. A minimum test-sample estimate of 1.32 is also attained at the same values of f and η. Figure 3 shows the piecewise-linear regression tree. Sample means and sizes are given beneath each terminal node. Since the major splits are on Year, it is easily seen from the node means that rate of incidence of mumps has in general decreased rapidly over time.

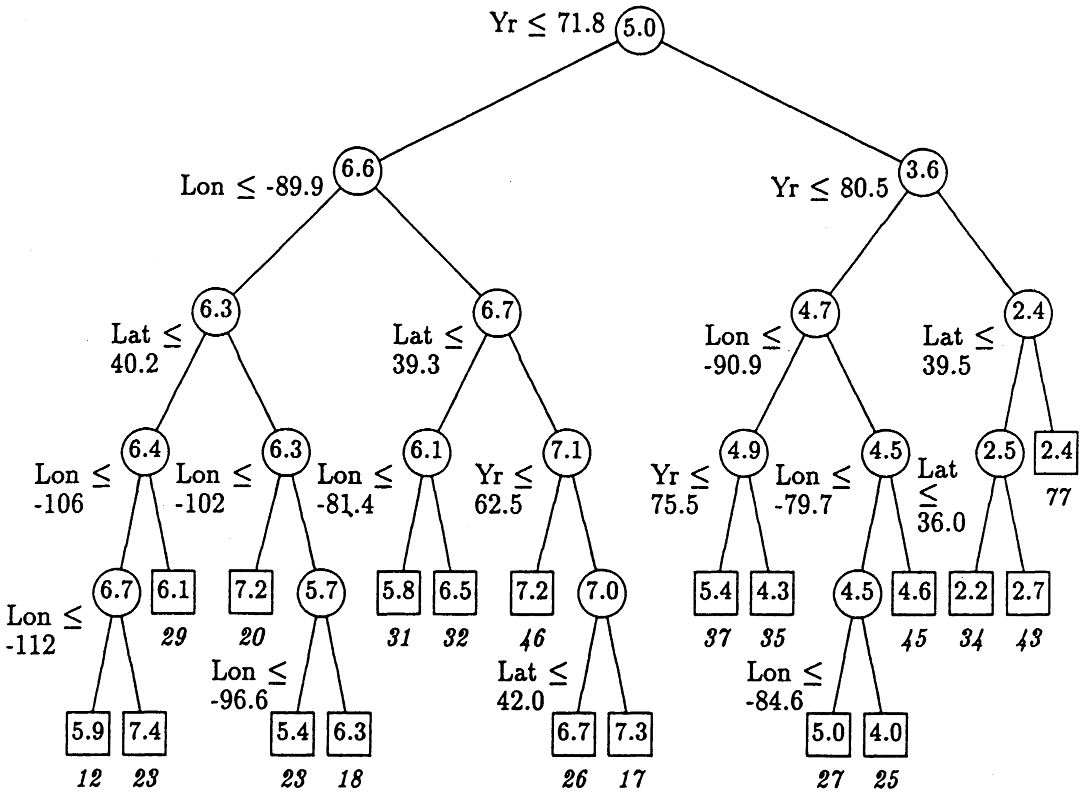

Figure 3. Piecewise-linear SUPPORT tree for log-rates of mumps based on 600 learning cases with $f = 0.1, \eta = 0.3$ and MINDAT = 30. The number in each node is the sample mean of the lograte per million population. Terminal node sample sizes are given in italics. The test-sample estimate of PMSE based on 923 test cases is 1.32.

Figure 4 shows the minimum cross-validation CART tree (called the "0-SE" tree in Breiman et al. (1984)). Its test-sample estimate of PMSE is 1.29. CART's default "1-SE" tree is shorter, with 15 terminal nodes, but its test-sample PMSE of 1.51 is worse. It is not easy to compare the SUPPORT and CART trees, except that SUPPORT is shorter and both trees split on Year first. Notice, however, that CART has a tendency to split a node into one large and one small subnode (Breiman et al. (1984) calls this "end-cut" preference), especially when there are outliers. This feature often produces very small terminal nodes (e.g., sample size 2 or 3) and big jumps in the function estimates. The SUPPORT method tends to prefer "middle-cuts," where the subnodes are roughly equal in size. Table 2 summarizes the test-sample estimates of PMSE of the methods, including the MARS and PIMPLE (Breiman (1991)) methods. They show that

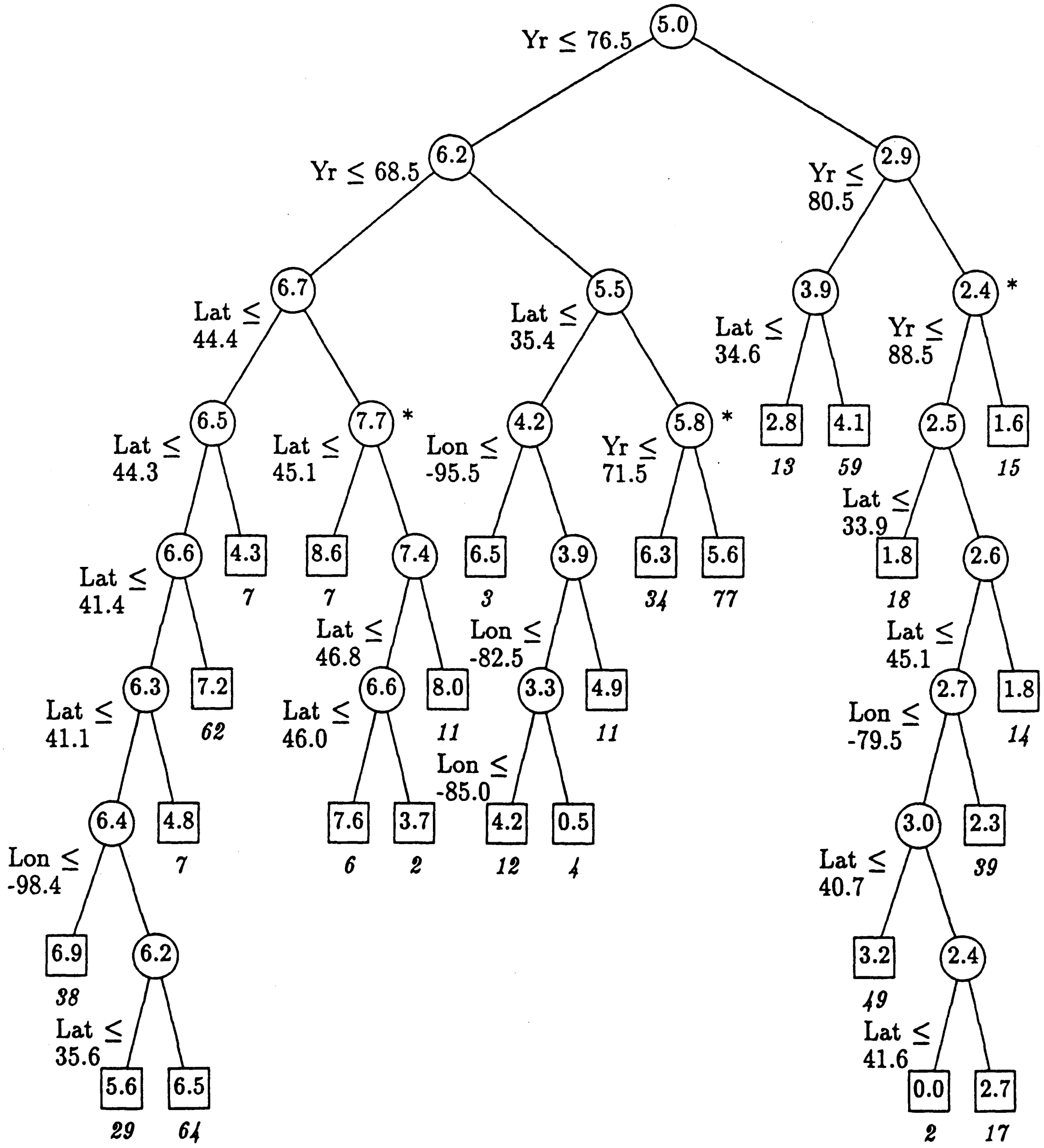

Figure 4. CART tree for mumps data based on the 0-SE rule. Nodes marked with asterisks are not split under the 1-SE rule. The number in each node is the sample mean of the lograte per million population. Terminal node sample sizes are given in italics. The test-sample estimates of PMSE are 1.29 and 1.51 for the 0-SE and 1-SE rules, respectively.

OLS is the worst. The CART and SUPPORT programs took 128 and 118 seconds, respectively, on a SUN 3/280 workstation to analyze this data set.

Table 2. Test-sample estimates of PMSE for mumps data. OLS denotes the ordinary least squares fit. The SUPPORT result is based on piecewise-linear fits without smoothing. The CART, MARS and PIMPLE results are obtained with their default tuning parameter values.

Method	OLS	SUPPORT	CART	CART	MARS	PIMPLE
			0-SE	1-SE		
PMSE	1.96	1.32	1.29	1.51	1.46	1.38

4.3. Baseball data: 132 learning cases and 16 regressors

Our next example is more challenging, because the data are very sparse — 132 cases in 17 dimensions. In addition, there are a couple of extreme outliers that have a dominating effect on the PMSE of the various function estimates. The data consist of information on the 1987 salaries and 16 other characteristics of 263 baseball hitters (the data file is also obtainable from the `statlib` archive). We take the dependent variable to be the natural logarithm of salary in thousands of dollars. As before, the data set was randomly divided into two parts, with 132 cases in the learning sample and 131 cases in the test sample. The regressor names and their regression coefficients from an OLS fit to the learning sample are given in the first three columns of Table 3. The R^2 value is quite low (62.8%) and no indication of lack of fit was evident from residual plots. Only Years and WalkC are significant, with the first having a positive coefficient and the second a negative coefficient.

The piecewise-linear SUPPORT tree is shown on the left of Figure 5. The same tree was obtained for all combinations of MINDAT $= 40, 50$ and $f, \eta = 0.1, 0.2, \ldots, 0.4$. The linear regression results for the two terminal nodes of the tree are given in the last four columns of Table 3. The variable Years is the only significant regressor in the right node. It is also significant in the left node, although two other regressors (Runs86 and RunbatC) are equally significant. The value of R^2 is increased substantially in one of the nodes and the residual standard deviation is reduced by thirty percent as a result of the split. Residual plots for both nodes are satisfactory. The fact that the signs for the Years coefficient are opposite for the two nodes suggests that Salary depends in a crucial way on this regressor.

In view of the sign-change in the Years variable in the subnodes, it may be thought that the addition of square terms to the ordinary regression model would produce a model competitive with the piecewise-linear one. The 32-term

Table 3. Estimated coefficients and t-ratios from linear regressions of log(salary) on 16 regressor variables for the root node and subnodes of the SUPPORT tree for the baseball data. Coefficients with absolute t-ratios greater than 2 are shown in italics.

		All years		Years ≤ 7		Years > 7	
Abbrev.	Regressor description	Coeff	t	Coeff	t	Coeff	t
Constant		*4.0729*	18.1	*4.2451*	20.8	*6.4355*	15.5
Tab86	#Times at bat in 1986	−0.0009	−0.5	−0.0001	−0.0	−0.0000	−0.0
Hits86	#Hits in 1986	0.0058	0.9	−0.0083	−1.2	0.0047	0.5
Home86	#Home runs in 1986	−0.0185	−1.1	−0.0189	−0.9	−0.0352	−1.6
Runs86	#Runs in 1986	0.0043	0.6	*0.0239*	2.7	0.0004	0.0
Runbat86	#Runs batted in in 1986	0.0091	1.3	0.0037	0.4	0.0107	1.3
Walk86	#Walks in 1986	0.0058	1.2	−0.0068	−1.1	−0.0016	−0.2
Years	#Years in major leagues	*0.0772*	2.3	*0.1107*	2.5	*−0.1331*	−3.1
TabC	#Times at bat during career	0.0006	1.7	0.0002	0.2	0.0001	0.3
HitsC	#Hits during career	−0.0031	−1.8	−0.0017	−0.4	−0.0004	−0.3
HomeC	#Home runs during career	−0.0043	−1.0	−0.0070	−0.9	−0.0015	−0.4
RunsC	#Runs during career	0.0035	1.7	0.0004	0.1	0.0010	0.6
RunbatC	#Runs batted in during career	0.0012	0.7	*0.0077*	2.4	0.0011	0.8
WalkC	#Walks during career	*−0.0019*	−2.2	0.0015	0.7	0.0001	0.1
Putout86	#Put outs in 1986	0.0002	0.7	−0.0004	−1.7	0.0001	0.4
Asst86	#Assists in 1986	−0.0005	−0.7	−0.0006	−0.8	−0.0006	−0.7
Err86	#Errors in 1986	0.0052	0.4	0.0048	0.4	0.0015	0.1
#Cases		132		82		50	
R^2		62.8%		84.3%		69.4%	
Resid. SD		0.59		0.39		0.41	

quadratic model does have an improved R^2 value of 85.6%, but it also has nine coefficients significant (absolute t-ratios greater than 3), thus making it harder to interpret. Further, its test-sample estimate of PMSE of 0.48 is larger than that of the piecewise-linear model because of two test cases that are badly predicted by the quadratic model.

The CART tree is shown on the right of Figure 5. The 0-SE tree has three terminal nodes and the 1-SE tree two terminal nodes. We observe that neither of the two variables (RunsC and RunbatC) appearing in the splits were found significant in the OLS fit to the whole learning sample. The SUPPORT tree, in contrast, splits on Years, the most important variable (as suggested by linear regression analysis as well as our knowledge of the sport). To check if this result is a fluke or not, we repeated the analysis switching the roles of the test sample and the learning sample. Using MINDAT = 40, SUPPORT produced the same

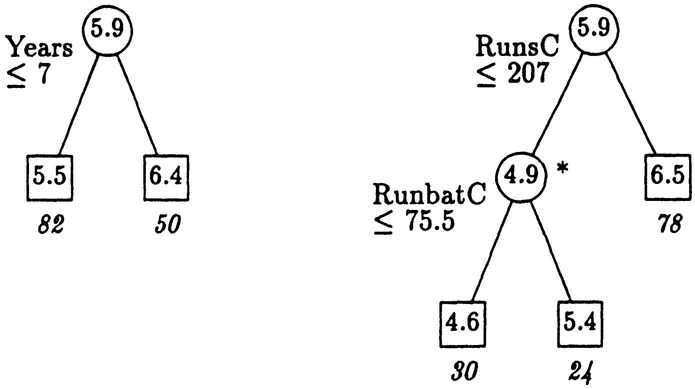

Figure 5. The tree on the left is the SUPPORT tree for the baseball player salary data constructed from the learning sample with $f = \eta = 0.3$ and MINDAT = 40. The 0-SE CART tree is on the right. The node with an asterisk is not split under the 1-SE rule. The number in each node is the sample mean of log(salary). Terminal node sample sizes are given in italics. The test-sample estimates of PMSE are 0.37 for SUPPORT and 0.42 and 0.49 for the 0-SE and 1-SE CART trees, respectively.

minimum cross-validation tree at $(f, \eta) = (0.2, 0.4)$, $(0.2, 0.5)$, $(0.3, 0.2)$, $(0.3, 0.3)$, $(0.3, 0.4)$, and $(0.4, 0.2)$. It is displayed on the left of Figure 6 and has the same structure as before. The minimum cross-validation CART tree on the right, however, looks quite different. The estimates of PMSE using the original learning sample as test cases were the same for CART and SUPPORT.

Finally, we combined the learning and test samples together to make one big learning sample and obtained the trees in Figure 7. The cross-validation estimate of PMSE of the SUPPORT tree was minimized at $(f, \eta) = (0.1, 0.4)$. The increase in data causes the SUPPORT tree to have one more split, with the first split the same as before. The CART tree has the same first split as that in Figure 6, although the other splits are quite different.

The relative instability of CART compared to SUPPORT is perhaps due to a fundamental difference between the two split selection strategies. CART selects the split that minimizes a weighted sum of the variances of its two subnodes, with the weights being the subnode sample sizes. Because the pool of candidate splits is often very large (approximately $131 \times 16 = 2096$ in the present example), the selected split can sometimes be highly affected by small local variations in the data values. On the other hand, SUPPORT picks the variable whose residual plot is most curved, as measured by Levene's test. Once this variable is chosen, the cut-point is fixed — it is the average of the means, along the selected variable axis, of two samples defined via the residuals. This means that a total of only 16 splits are considered in the present example. Another consequence of the

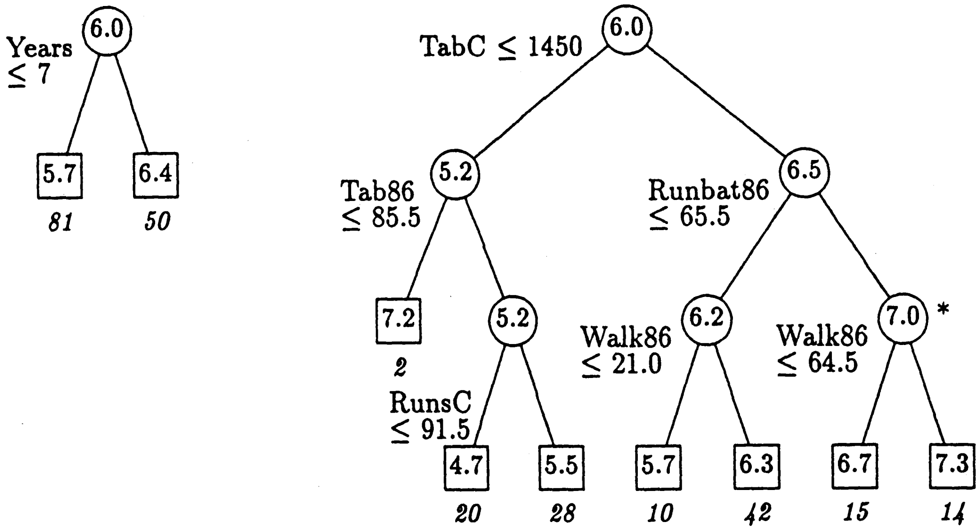

Figure 6. Regression trees for the baseball player salary data constructed using the 131 test cases as learning sample. The SUPPORT tree on the left is similar to the one in Figure 5. The tree on the right is obtained with CART's 0-SE rule; the node with an asterisk is not split under the 1-SE rule. The number in each node is the sample mean of log(salary). Node sample sizes are given in italics.

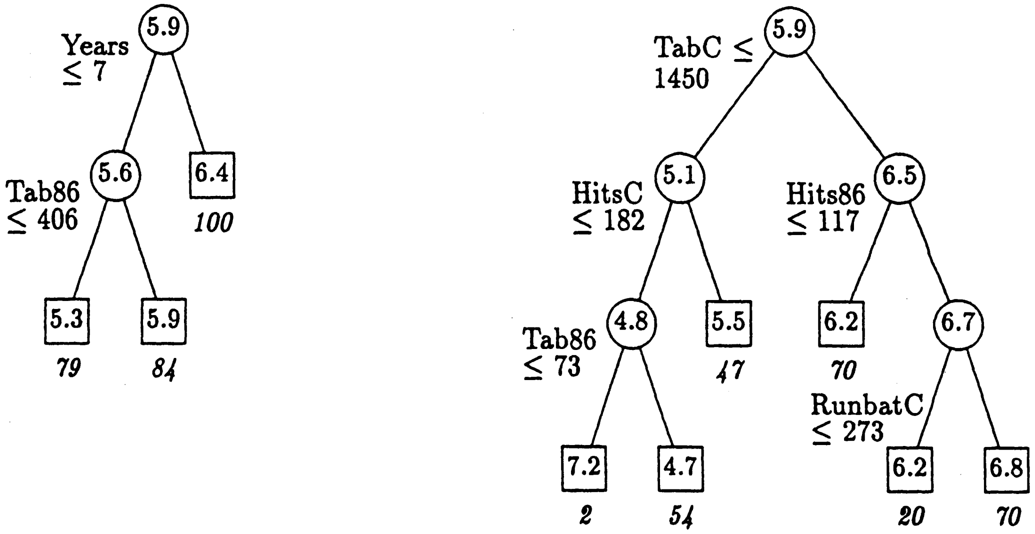

Figure 7. Regression trees for the baseball player salary data constructed using all 263 cases (learning and test samples combined). The SUPPORT tree is on the left and the 0-SE CART tree is on the right (the 1-SE tree is the same). The number in each node is the sample mean of log(salary). The sample size is given in italics beneath each terminal node.

difference in strategies is that SUPPORT tends to produce "middle-cuts" (where the sample sizes of the subnodes are roughly equal) whereas CART sometimes prefers "end-cuts" (unbalanced sample sizes in its subnodes) especially when outliers are present. The CART and SUPPORT programs took 85 and 50 seconds, respectively, on a SUN 3/280 workstation to analyze this data set.

Table 4 gives the various test-sample estimates of PMSE. The PIMPLE method is the worst. Repeating the analysis with different random partitions of the data into learning and test samples did not produce any significant improvement in the test-sample estimate of PMSE for PIMPLE.

Table 4. Test-sample estimates of PMSE for baseball player salary data based on 131 test samples. Function estimates are based on 132 learning samples. The piecewise-linear regression method with MINDAT equal to 40 or 50 gives the same tree.

Method	Linear regression	Quadratic regression	SUPPORT	CART 0-SE	CART 1-SE	MARS	PIMPLE
PMSE	0.53	0.48	0.37	0.42	0.49	0.36	7.7

5. Large-Sample Results

We now investigate the large-sample behavior of the piecewise-polynomial and the smooth estimates. Let $(Y_1, \boldsymbol{X}_1), (Y_2, \boldsymbol{X}_2), \ldots, (Y_n, \boldsymbol{X}_n)$ be n independent observations forming the learning sample. Here the Y_i's are real-valued random variables and the $\boldsymbol{X}_i$'s take their values in the K-dimensional Euclidean space R^K. The $\boldsymbol{X}_i$'s may be random or deterministic depending on the situation. Let C be a compact (closed and bounded) set with a nonempty interior contained in the range of the $\boldsymbol{X}_i$'s. We want to estimate the function g over C. Let T_n be a "cover" (possibly random) of C based on the learning sample. That is, T_n is a collection of subsets of the regressor space such that $C = \cup_{t \in T_n} t$. Note that T_n may or may not be a partition (some of the sets in T_n may have nonempty intersections). For example, the recursive partitioning algorithm will yield a collection of disjoint sets, but if we work with extended nodes (e.g., t_τ with $\tau > 0$ as discussed in Section 3 and demonstrated in Example 4.1), we get overlapping sets. We will assume that any set in T_n is a polyhedron in R^K having at most M faces, where M is a fixed positive integer (Breiman et al. (1984, p.319)). For $t \in T_n$, we write $\delta(t)$ to denote the diameter of t defined as the supremum of all possible values of $|\boldsymbol{x} - \boldsymbol{y}|$ as $\boldsymbol{x}$ and $\boldsymbol{y}$ vary over the set t. We define $|T_n|$, the norm of the collection T_n, as $|T_n| = \max_{t \in T_n} \delta(t)$.

For any $t \in T_n$, let N_t and $\bar{\boldsymbol{X}}_t$ denote the number and the average of the $\boldsymbol{X}_i$'s that belong to t. So, $N_t = \#\{\boldsymbol{X}_i | \boldsymbol{X}_i \in t\}$ and $\bar{\boldsymbol{X}}_t = N_t^{-1} \sum_{\boldsymbol{X}_i \in t} \boldsymbol{X}_i$. Assuming that the regression function $g(\boldsymbol{x}) = E(Y | \boldsymbol{X} = \boldsymbol{x})$ is mth order differentiable

($m \geq 0$), the Taylor expansion of g around $\bar{\boldsymbol{X}}_t$ is given as

$$g(\boldsymbol{x}) = \sum_{\boldsymbol{u} \in \boldsymbol{U}} (\boldsymbol{u}!)^{-1} D^{\boldsymbol{u}} g(\bar{\boldsymbol{X}}_t)(\boldsymbol{x} - \bar{\boldsymbol{X}}_t)^{\boldsymbol{u}} + r_t(\boldsymbol{x}, \bar{\boldsymbol{X}}_t).$$

Here $r_t(\boldsymbol{x}, \bar{\boldsymbol{X}}_t)$ is the remainder term and, for $\boldsymbol{u} \in \boldsymbol{U}$ and $\boldsymbol{x} \in R^K$, we define $\boldsymbol{u}! = \prod_{i=1}^K u_i!$ and $\boldsymbol{x}^{\boldsymbol{u}} = \prod_{i=1}^K x_i^{u_i}$ with the convention that $0! = 0^0 = 1$. Let $s(\boldsymbol{U}) = \#(\boldsymbol{U})$ and $\boldsymbol{\beta} = (b_{\boldsymbol{u}})_{\boldsymbol{u} \in U}$ be a vector of dimension $s(\boldsymbol{U})$. We write $P(\boldsymbol{x}, \boldsymbol{\beta}, \bar{\boldsymbol{X}}_t)$ to denote the polynomial (in $\boldsymbol{x}$)

$$P(\boldsymbol{x}, \boldsymbol{\beta}, \bar{\boldsymbol{X}}_t) = \sum_{\boldsymbol{u} \in \boldsymbol{U}} b_{\boldsymbol{u}} (\boldsymbol{u}!)^{-1} \{\delta(t)\}^{-[\boldsymbol{u}]} (\boldsymbol{x} - \bar{\boldsymbol{X}}_t)^{\boldsymbol{u}}.$$

Let $\hat{g}_n$ denote the estimate of g constructed from the learning sample by piecewise-polynomial least squares fit. In other words, $\hat{g}_n$ is obtained by fitting polynomials to different subsets (corresponding to different sets in the collection T_n) of the learning sample by solving the following minimization problem for every $t \in T_n$:

$$\text{Minimize } \sum_{\boldsymbol{X}_i \in t} \{Y_i - P(\boldsymbol{X}_i, \boldsymbol{\beta}, \bar{\boldsymbol{X}}_t)\}^2 \text{ with respect to } \boldsymbol{\beta} \in R^{s(\boldsymbol{U})}. \quad (5)$$

So, if $\hat{\boldsymbol{\beta}}_t$ is a solution to (5), we have $\hat{g}_n(\boldsymbol{x}) = P(\boldsymbol{x}, \hat{\boldsymbol{\beta}}_t, \bar{\boldsymbol{X}}_t)$ for any $\boldsymbol{x} \in t$. Further, we can use $D^{\boldsymbol{u}} P(\boldsymbol{x}, \hat{\boldsymbol{\beta}}_t, \bar{\boldsymbol{X}}_t)$ as an estimate of $D^{\boldsymbol{u}} g(\boldsymbol{x})$ for $\boldsymbol{u} \in \boldsymbol{U}$ and $\boldsymbol{x} \in t$. If the sets in T_n are overlapping, $\boldsymbol{x}$ may belong to more than one of the sets in T_n. In that case one can use a simple and convenient rule (e.g., average of different values) so that $\hat{g}_n$ and various derivative estimates are well defined.

At this point, we state a set of sufficient conditions that guarantee a very desirable asymptotic property of the piecewise-polynomial estimate of the regression function and the associated derivative estimates. In practice, the behavior of the estimates will largely depend on the choice of the tuning parameters (e.g., f and η). Nevertheless, the following technical conditions and the theorems provide useful insights into the large-sample properties of the proposed estimates.

For $\boldsymbol{X}_i \in t$, let Γ_i be the $s(\boldsymbol{U})$-dimensional column vector with components given by $(\boldsymbol{u}!)^{-1}\{\delta(t)\}^{-[\boldsymbol{u}]}(\boldsymbol{X}_i - \bar{\boldsymbol{X}}_t)^{\boldsymbol{u}}$, where $\boldsymbol{u} \in \boldsymbol{U}$. We denote by D_t the $s(\boldsymbol{U}) \times s(\boldsymbol{U})$ matrix defined as $\sum_{\boldsymbol{X}_i \in t} \Gamma_i \Gamma_i^T$, where T indicates transpose. Then, it is obvious that whenever D_t is nonsingular, $\hat{\boldsymbol{\beta}}_t = D_t^{-1} \sum_{\boldsymbol{X}_i \in t} \Gamma_i Y_i$.

Condition (a): $\max_{t \in T_n} \sup_{\boldsymbol{x} \in t} \{\delta(t)\}^{-m} |r_t(\boldsymbol{x}, \bar{\boldsymbol{X}}_t)| \xrightarrow{P} 0$ as $n \to \infty$.

Condition (b): Let $G_n = \min_{t \in T_n} \{\delta(t)\}^{2m} N_t$. Then $G_n / \log n \xrightarrow{P} \infty$ as $n \to \infty$.

Condition (c): Let λ_t be the smallest eigenvalue of $N_t^{-1} D_t$ and $\lambda_n = \min_{t \in T_n} \lambda_t$. Then λ_n remains bounded away from 0 in probability as $n \to \infty$.

Condition (d): Let $\psi(a|\boldsymbol{x}) = E(e^{a|Y-g(\boldsymbol{x})|}|\boldsymbol{X} = \boldsymbol{x})$. Then there is $a > 0$ such that $\psi(a|\boldsymbol{x})$ is bounded on C.

Theorem 1. *Suppose that conditions* (a) *through* (d) *hold. Then for every* $\boldsymbol{u} \in \boldsymbol{U}$,

$$\max_{t \in T_n} \sup_{\boldsymbol{x} \in t} |D^{\boldsymbol{u}} P(\boldsymbol{x}, \hat{\boldsymbol{\beta}}_t, \bar{\boldsymbol{X}}_t) - D^{\boldsymbol{u}} g(\boldsymbol{x})| \xrightarrow{P} 0 \quad \textit{as} \quad n \to \infty.$$

The theorem is proved in the Appendix. Note that, because of the smoothness assumptions on g, Condition (a) will be satisfied whenever $|T_n| \xrightarrow{P} 0$ as $n \to \infty$ (cf. condition (12.9) in Breiman et al. (1984), and the assumptions in their Theorem 12.13). On the other hand, it is not necessary that $|T_n| \xrightarrow{P} 0$ in order for Condition (a) to be satisfied. For example, if $g(\boldsymbol{x})$ is a polynomial or very close to a polynomial in $\boldsymbol{x}$ over certain regions of the regressor space, then some of the sets in the collection T_n may not shrink in size as n increases. Note also that Condition (d), which is a moment condition imposed on the random error ε, is the same as condition (12.12) of Breiman et al. (1984). Condition (b) ensures that there will be sufficiently many data points in each terminal node as the total sample size grows. Condition (c) is needed because we are fitting a polynomial to the data points corresponding to each set in T_n instead of the histogram-type piecewise constant estimate of Breiman et al. (1984). It guarantees that in a large sample, each of the matrices D_t's for $t \in T_n$ will be nonsingular with high probability.

The theorem asserts that our preliminary estimates of the regression function and its various derivatives are *consistent* provided Conditions (a) through (d) hold. However, these preliminary estimates are not smooth everywhere because they are piecewise-polynomials. The following theorem, which is an immediate consequence of the theorem above, asserts that if we use an appropriately chosen "smooth partition of unity" (see, e.g., Rudin (1973)) on C to smooth our preliminary estimates, we get *smooth and consistent* estimates for the regression function and its derivatives.

Theorem 2. *For* $n \geq 1$, *let* T_n^* *be a partition* (*possibly random*) *that refines the collection* T_n (*i.e., for any* $t^* \in T_n^*$, *there is a* $t \in T_n$ *such that* $t^* \subseteq t$), *and let* $\{W_n(\boldsymbol{x}, t^*)\}_{t^* \in T_n^*}$ *be a collection of smooth nonnegative functions on* C *with the following properties.*

1. $\sum_{t^* \in T_n^*} W_n(\boldsymbol{x}, t^*) = 1$ *for all* $\boldsymbol{x} \in C$ *and* $n \geq 1$.

2. *For any* $\boldsymbol{x} \in C$, *let* $S_n(\boldsymbol{x})$ *be the union of the sets* $t^* \in T_n^*$ *such that* $W_n(\boldsymbol{x}, t^*) > 0$, *i.e.,* $S_n(\boldsymbol{x}) = \{\boldsymbol{y} | \boldsymbol{y} \in t^*$ *and* $W_n(\boldsymbol{x}, t^*) > 0\}$. *Let* $|S_n(\boldsymbol{x})|$ *denote the diameter of this set. Then* $\boldsymbol{x} \in S_n(\boldsymbol{x})$ *and* $\sup_{\boldsymbol{x} \in C} |S_n(\boldsymbol{x})| \xrightarrow{P} 0$ *as* $n \to \infty$.

Let $\hat{g}_{t^*}$ be the piece of $\hat{g}_n$ that is defined on $t^* \in T_n^*$. So, $\hat{g}_{t^*}(\boldsymbol{x}) = \hat{g}_n(\boldsymbol{x})$ if $\boldsymbol{x} \in t^*$. We extend $\hat{g}_{t^*}$ on the entire R^K space using its natural smooth extension. Define $D^{\boldsymbol{u}} g_n^*(\boldsymbol{x}) = \sum_{t^* \in T_n^*} W_n(\boldsymbol{x}, t^*) D^{\boldsymbol{u}} \hat{g}_{t^*}(\boldsymbol{x})$, where $\boldsymbol{x} \in C$, and assume that Conditions (a) *through* (d) *hold. Then, for every $\boldsymbol{u} \in \boldsymbol{U}$, $\sup_{\boldsymbol{x} \in C} |D^{\boldsymbol{u}} g_n^*(\boldsymbol{x}) - D^{\boldsymbol{u}} g(\boldsymbol{x})| \xrightarrow{P} 0$ as $n \to \infty$.*

The proof follows immediately from Theorem 1, conditions (1) and (2) of the theorem, and the smoothness of g on the compact set C. Note that a collection of functions $W_n(\boldsymbol{x}, t^*)$ satisfying conditions (1) and (2) of the theorem exists in all the situations that typically arise in practice. For example, if C is a rectangle in R^K and T_n^* is a collection of rectangles, one can easily construct such collections using B-splines (see, e.g., de Boor (1978, Ch.9)) or tensor products of B-splines whenever $|T_n^*| \xrightarrow{P} 0$ as $n \to \infty$. For a general theorem on the existence of a "locally finite and smooth partition of unity" on an arbitrary domain, see Chapter 6 of Rudin (1973). As we have noted before, if the collection T_n is obtained via a recursive partitioning algorithm, $|T_n|$ may not converge to 0 in probability as n increases in some cases. However, one can construct T_n^* from T_n by splitting each of the "big" sets in T_n into smaller ones, keeping $\hat{g}_n$ unchanged.

6. Concluding Remarks

The motivation for this research was to find new techniques for tree-structured regression and to compare them with CART and some recent spline-based methods. Along the way, the following specific goals became apparent and solutions were found for them.

1. Use of piecewise-polynomial fits instead of piecewise-constant fits. This has two important advantages. First, a piecewise-polynomial tree typically has fewer splits than its piecewise-constant counterpart. This aids interpretation. Second, the vast accumulated pool of knowledge and experience with ordinary linear regression can be used to advantage in the analysis of piecewise-linear regression trees.

2. Substitution of CART's intensive search method of split selection with a faster method based on residual analysis. As shown in the last example, our solution tends to choose variables that are more indicative of global trends. It also yields tree structures that are more robust against small perturbations in the data.

3. Replacement of CART's cross-validation pruning with a more natural multi-step look-ahead stopping rule. Direct stopping rules have been criticized in the literature for good reason, because they have traditionally been too naive. By endowing a direct stopping rule with multi-step look-ahead and the power of cross-validation, we believe that we have found a viable alternative.

4. Simple and efficient smoothing of the piecewise-polynomial estimate. Many real applications require smooth estimates because the true regression function is known to be smooth. Our search for a solution had to overcome two constraints. First, because recursive partitioning methods are already compute-intensive, the additional smoothing must be done in an inexpensive way. Second, the properties of the smoothed estimate should be easy to analyze. The method of weighted averaging achieves these goals. An unexpected additional benefit is the use of weighted averages of the derivatives of the piecewise-polynomials to estimate the corresponding derivatives of the function.

5. Asymptotic consistency of our estimate. Although the results reported here are quite general and do not specifically apply to a particular tree-structured regression procedure, they do extend our understanding of the types of situations in which SUPPORT would yield reasonable results. The main difficulty lies in our currently incomplete understanding of the properties of the cross-validatory techniques used in SUPPORT.

It is interesting to note from the examples that, as far as PMSE is concerned, the SUPPORT method is competitive against modern and well-regarded non-tree regression methods such as MARS and PIMPLE. The main practical advantages of SUPPORT are: (i) its solutions can be represented as decision trees, and (ii) the piecewise estimates have familiar forms, such as linear or quadratic polynomials. Analytic tractability of the large-sample properties of the estimates is another advantage not shared by MARS or PIMPLE.

The FORTRAN code for the SUPPORT program may be obtained from W.-Y. Loh.

Appendix: Proof of Theorem 1

Observe that the assertion in the Theorem will follow if we can show that as $n \longrightarrow \infty$,

$$\max_{t \in T_n} \{\delta(t)\}^{-m} \left| \hat{\boldsymbol{\beta}}_t - \Delta(\bar{\boldsymbol{X}}_t) \right| \xrightarrow{P} 0,$$

where $\Delta(\bar{\boldsymbol{X}}_t)$ denotes the $s(\boldsymbol{U})$-dimensional vector whose components are $D^{\boldsymbol{u}}$ $g(\bar{\boldsymbol{X}}_t)\{\delta(t)\}^{[\boldsymbol{u}]}$ with $\boldsymbol{u} \in \boldsymbol{U}$. Condition (c) ensures that, with probability tending to 1 as $n \rightarrow \infty$, the matrices D_t's for $t \in T_n$ are all nonsingular. Hence, writing $Y_i = g(\boldsymbol{X}_i) + \varepsilon_i$ so that $E(\varepsilon_i | \boldsymbol{X}_i) = 0$, we have

$$\begin{aligned} \hat{\boldsymbol{\beta}}_t &= D_t^{-1} \sum_{\boldsymbol{X}_i \in t} \Gamma_i Y_i \\ &= D_t^{-1} \sum_{\boldsymbol{X}_i \in t} \Gamma_i g(\boldsymbol{X}_i) + D_t^{-1} \sum_{\boldsymbol{X}_i \in t} \Gamma_i \varepsilon_i, \qquad (6) \end{aligned}$$

for all $t \in T_n$ on a set that has probability tending to 1 as $n \to \infty$. By straightforward algebra using the Taylor expansion of g around $\bar{\boldsymbol{X}}_t$, we can rewrite (6) as

$$\hat{\boldsymbol{\beta}}_t - \Delta(\bar{\boldsymbol{X}}_t) = D_t^{-1} \sum_{\boldsymbol{X}_i \in t} \Gamma_i r_t(\boldsymbol{X}_i, \bar{\boldsymbol{X}}_t) + D_t^{-1} \sum_{i \in t} \Gamma_i \varepsilon_i.$$

Denote by B_t the first term and by V_t the second term on the right above. The term B_t can be thought of as a "bias" term due to the polynomial approximation of the function g inside the set t, and the term V_t as the "variance" term caused by the random noise ε_i. Such a decomposition is typical and arises quite naturally in almost every nonparametric regression analysis (see, e.g., Stone (1980, 1982)). From Conditions (a) and (c), we conclude that $\max_{t \in T_n} |\{\delta(t)\}^{-m} B_t| \xrightarrow{P} 0$ as $n \to \infty$.

Note that ε_i is a random variable whose conditional mean given $\boldsymbol{X}_i$ is 0 and that Condition (d) implies the existence of constants $k_1 > 0$ and $r > 0$ such that $\psi(w|\boldsymbol{x}) \leq 2\exp\{k_1 w^2/2\}$ for all $\boldsymbol{x} \in C$ and $0 \leq w \leq r$ (see the arguments at the beginning of Lemma 12.27 in Breiman et al. (1984)). Let us now pretend that t is a fixed non-random polyhedron in R^K, all the data points $\boldsymbol{X}_i$'s that fall in t are a set of fixed deterministic points in C, and the corresponding ε_i's form a set of independent random variables such that the distribution of ε_i is the same as the conditional distribution of ε_i given $\boldsymbol{X}_i$ in the original sample. Then, an application of Lemma 12.26 in Breiman et al. (1984) to each component of the $s(\boldsymbol{U})$-dimensional vector $\{\delta(t)\}^{-m} N_t^{-1} \sum_{\boldsymbol{X}_i \in t} \Gamma_i \varepsilon_i$ implies that there exist constants $k_2 > 0$, $k_3 > 0$ and $\omega_0 > 0$ (which depend only on the compact set C, the integer $s(\boldsymbol{U})$ and the constant k_1) such that

$$\Pr\left(\{\delta(t)\}^{-m} \left| N_t^{-1} \sum_{\boldsymbol{X}_i \in t} \Gamma_i \varepsilon_i \right| > \omega\right) \leq k_2 \exp\{-k_3 \{\delta(t)\}^{2m} N_t \omega^2\},$$

whenever $\omega \leq \omega_0$. Finally, using arguments that are essentially identical to those in the proof of Lemma 12.27 in Breiman et al. (1984), by exploiting Conditions (b) and (c) and the fundamental combinatorial result of Vapnik and Chervonenkis (1971) (see the proof of Lemma 12.23 in Breiman et al. (1984, p.330)), we conclude that $\max_{t \in T_n} |\{\delta(t)\}^{-m} V_t| \xrightarrow{P} 0$ as $n \to \infty$.

Acknowledgement

Chaudhuri's research was partially supported by funds from the University of Wisconsin Graduate School. Loh's research was partially supported by NSF grant DMS 88-03271 and ARO grant DAAL03-91-G-0111.

References

Ahn, H. and Loh, W.-Y. (1994). Tree-structured proportional hazards regression modeling. *Biometrics.* In press.

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988). *The New S Language.* Wadsworth, Pacific Grove.

Breiman, L. (1991). The Π method for estimating multivariate functions from noisy data (with discussion). *Technometrics* **33**, 125–160.

Breiman, L. and Meisel, W. S. (1976). General estimates of the intrinsic variability of data in nonlinear regression models. *J. Amer. Statist. Assoc.* **71**, 301–307.

Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J. (1984). *Classification and Regression Trees.* Wadsworth, Belmont.

Buja, A., Hastie, T. and Tibshirani, R. (1989). Linear smoothers and additive models (with discussion). *Ann. Statist.* **17**, 453–555.

Cleveland, W. S. and Devlin, S. J. (1988). Locally weighted regression: An approach to regression analysis by local fitting. *J. Amer. Statist. Assoc.* **83**, 596–610.

de Boor, C. (1978). *A Practical Guide to Splines.* Springer, New York.

Eubank, R. L. (1988). *Spline Smoothing and Nonparametric Regression.* Dekker, New York.

Friedman, J. H. (1979). A tree-structured approach to nonparametric multiple regression. *Smoothing Techniques for Curve Estimation* (Edited by T. Gasser and M. Rosenblatt), 5–22. *Lecture Notes in Math.* **757**, Springer-Verlag, New York.

Friedman, J. H. (1991). Multivariate adaptive regression splines (with discussion). *Ann. Statist.* **19**, 1–141.

Friedman, J. H. and Stuetzle, W. (1981). Projection pursuit regression. *J. Amer. Statist. Assoc.* **76**, 817–823.

Gasser, T. and Muller, H.-G. (1979). *Kernel Estimation of Regression Functions. Lecture Notes in Math.* **757**, 23–68, Springer-Verlag, New York.

Gasser, T. and Muller, H.-G. (1984). Estimating regression functions and their derivatives by the kernel method. *Scand. J. Statist.* **11**, 171–185.

Hastie, T. and Tibshirani, R. (1990). *Generalized Additive Models.* Chapman and Hall, London.

Huang, M.-C. (1989). Piecewise-linear tree-structured regression. PhD thesis, University of Wisconsin, Madison, Department of Statistics.

Levene, H. (1960). Robust tests for equality of variances. *Contributions to Probability and Statistics* (Edited by I. Olkin, S. G. Ghurye, W. Hoeffding, W. G. Madow and H. B. Mann), 278–292, Stanford University Press, London.

Loh, W.-Y. (1991). Survival modeling through recursive stratification. *Comput. Statist. Data Anal.* **12**, 295–313.

Loh, W.-Y. and Vanichsetakul, N. (1988). Tree-structured classification via generalized discriminant analysis (with discussion). *J. Amer. Statist. Assoc.* **83**, 715–728.

Nadarya, E. A. (1964). On estimating regression. *Theory Probab. Appl.* **9**, 141–142.

O'Sullivan, F. (1991). Discussion of 'Multivariate adaptive regression splines' by J. H. Friedman. *Ann. Statist.* **19**, 99–102.

Ramsay, J. O. (1988). Monotone regression splines in action (with discussion). *Statist. Sci.* **3**, 425–461.

Rosenblatt, M. (1971). Curve estimates. *Ann. Math. Statist.* **42**, 1815–1842.

Rudin, W. (1973). *Functional Analysis.* McGraw-Hill, New York.

Sonquist, J. (1970). Multivariate model building: The validation of a search strategy. Technical Report, Institute for Social Research, University of Michigan, Ann Arbor.

Stone, C. J. (1980). Optimal rates of convergence for nonparametric estimators. *Ann. Statist.* **8**, 1348–1360.

Stone, C. J. (1982). Optimal global rates of convergence for nonparametric regression. *Ann. Statist.* **10**, 1040–1053.

Stone, C. J. (1985). Additive regression and other nonparametric models. *Ann. Statist.* **13**, 689–705.

Vapnik, V. N. and Chervonenkis, A. Y. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory Probab. Appl.* **16**, 264–280.

Wahba, G. (1990). *Spline Models for Observational Data.* CBMS-NSF Regional Conference Series in Applied Mathematics, SIAM, Philadelphia.

Watson, G. S. (1964). Smooth regression analysis. *Sankhyā Ser.A* **26**, 359–372.

Division of Theoretical Statistics & Mathematics, Indian Statistical Institute, Calcutta 700035, India.
Department of Statistics, National Cheng Kung University, Tainan, Taiwan.
Department of Statistics, University of Wisconsin, Madison, WI 53706, U.S.A.

(Received February 1992; accepted September 1993)