Bump Hunting in High–Dimensional Data

Jerome H. Friedman* & Nicholas I. Fisher†

October 28, 1998

Abstract

Many data analytic questions can be formulated as (noisy) optimization problems. They explicitly or implicitly involve finding simultaneous combinations of values for a set of ("input") variables that imply unusually large (or small) values of another designated ("output") variable. Specifically, one seeks a set of subregions of the input variable space within which the value of the output variable is considerably larger (or smaller) than its average value over the entire input domain. In addition it is usually desired that these regions be describable in an interpretable form involving simple statements ("rules") concerning the input values. This paper presents a procedure directed towards this goal based on the notion of "patient" rule induction. This patient strategy is contrasted with the greedy ones used by most rule induction methods, and semi-greedy ones used by some partitioning tree techniques such as CART. Applications involving scientific and commercial data bases are presented.

Keywords: Data Mining, noisy function optimization, classification, association, rule induction.

1. Introduction

The purpose of many data analyses can be viewed in the context of "prediction". The data base contains repeated observations of a designated "output" variable y along with simultaneous values of additional "input" variables $\mathbf{x} = (x_1, x_2, \cdots, x_n)$. The goal is to use these data

$$\{y_i, \mathbf{x}_i\}_1^N \tag{1.1}$$

to determine likely values of y for specified (future) values of the inputs $\mathbf{x}$. Supposing the data (1.1) is a random sample from some (unknown) joint distribution with probability density $p(y, \mathbf{x})$, this goal can be characterized as trying to obtain the probability density of y-values at each $\mathbf{x}$

$$p(y \,|\, \mathbf{x}) = \frac{p(y, \mathbf{x})}{\int p(y, \mathbf{x})\, dy}$$

which is most simply (if incompletely) described by its first moment

$$f(\mathbf{x}) = E[y \,|\, \mathbf{x}] = \int y\, p(y \,|\, \mathbf{x})\, dy. \tag{1.2}$$

This quantity (1.2) is well known to minimize mean-squared prediction error at each $\mathbf{x}$

$$f(\mathbf{x}) = \arg\min_f E[(y - f)^2 | \, \mathbf{x}].$$

*Department of Statistics and Stanford Linear Accelerator Center, Stanford University, Stanford, CA 94305. (jhf@stat.stanford.edu)

†CSIRO Mathematical & Information Sciences, Locked Bag 17, North Ryde, NSW 2113, Australia. (Nick.Fisher@cmis.CSIRO.AU)

Expressing the output variable as

$$y = E[y \,|\, \mathbf{x}] + (y - E[y \,|\, \mathbf{x}]) = f(\mathbf{x}) + \varepsilon \tag{1.3}$$

one sees that this prediction problem can be cast as one of function estimation where the goal is to approximate ("learn") a deterministic "target" function $f(\mathbf{x})$ from a set of observations where its value (at each $\mathbf{x}$) is corrupted by random noise ε. This noise represents the random distribution of the output y about its mean value $f(\mathbf{x})$, at each $\mathbf{x}$, and characterizes the fact that specifying a simultaneous set of input values does not specify a unique y-value; other factors, not captured by the set of measured inputs, influence the output value.

Although restricting attention to the first moment (1.2) greatly simplifies the problem, it is still a formidable one. The problem of accurately approximating a general function of many arguments, everywhere within some domain of input values, based on sampled data (with or without noise) remains a difficult one. As such, it has been the focus of much research in the fields of mathematics, neural networks, machine learning, and statistics [see Lorentz (1986), Bishop (1995), Ripley (1996), Mitchell (1997), and Wahba (1990)]. The set of functions amenable to accurate approximation with current methodology is still relatively small and there may well be many target functions to be encountered in practice that remain elusive.

Often, function approximation is applied in situations for which the actual data analytic goal is far more modest; the interest is in some *property* of the target function. A common procedure in such situations is to attempt to estimate the target $f(\mathbf{x})$ everywhere in the input space and ascertain the property of interest from the resulting estimate $\hat{f}(\mathbf{x})$. Frequently however, this strategy can be counter–productive in that an alternative one focused directly on estimating the property of interest may give rise to higher accuracy.

One example of this phenomenon is (2 - class) classification. Here the output variable y assumes two values, $y = 0$ indicating the first class and $y = 1$ indicating the second. The target function is

$$f(\mathbf{x}) = E[y \,|\, \mathbf{x}] = \Pr(y = 1 \,|\, \mathbf{x}) \tag{1.4}$$

and the optimal (minimum error) classification decision is

$$y = 1(f(\mathbf{x}) > 1/2), \tag{1.5}$$

where $1(\cdot)$ takes the value one when its argument is true and zero otherwise. It is common to apply function approximation methodology in this situation, inserting the estimate $\hat{f}(\mathbf{x})$ in (1.5) to make predictions $\hat{y}$. Vapnik (1995) and Friedman (1997) have shown that this can provide poor performance relative to procedures that try to directly estimate the decision boundary $f(\mathbf{x}) = 1/2$, and sometimes leads to counter–intuitive results. For example improving the quality of the function estimate can actually degrade classification performance.

Another example is (univariate) density estimation where the target $f(x)$ is the relative probability of an observation at x. The property of interest is the cumulative distribution function

$$F(x) = \int_{-\infty}^{x} f(x')\; dx'. \tag{1.6}$$

Inserting the optimal kernel density estimate of $f(x)$ in (1.6) leads to a lower accuracy estimate of $F(x)$ than simply using the raw data as the density estimate

$$\hat{F}(x) = \frac{1}{N} \sum_{i=1}^{N} 1(x_i \leq x)$$

[Hall (1989)]. The raw data is of course a very much poorer estimate of the density $f(x)$ itself.

Another (trivial) example is when the property of interest is the mean of the target function over the entire input space

$$\bar{f} = \int f(\mathbf{x})\, p(\mathbf{x})\, d\mathbf{x}. \tag{1.7}$$

Here the sample output mean

$$\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{1.8}$$

is likely to provide higher accuracy than taking the mean of some estimate $\hat{f}(\mathbf{x})$ derived from the data.

2. Function optimization

Function approximation often is applied in contexts where the property of interest is the optimal (maximum or minimum) values of the target $f(\mathbf{x})$ (1.2). Specifically, one seeks a subregion of the space of input values within which the average value of target is much larger (or smaller) that its average over the entire input space (1.7). Since minimizing a function can be achieved by maximizing its negative, only maximization is considered here without loss of generality. Let S_j be the set of all possible values for the input variable x_j

$$\{x_j \in S_j\}_{j=1}^{n}. \tag{2.1}$$

The individual S_j could represent real (perhaps discrete) values, or categorical (unorderable) values. The entire input domain S can then be represented by the n-dimensional (outer) product space

$$S = S_1 \times S_2 \times \cdots \times S_n. \tag{2.2}$$

The goal is to find a subregion R of the input domain S, $R \subset S$, for which

$$\bar{f}_R = ave_{\mathbf{x}\in R}\, f(\mathbf{x}) = \int_{\mathbf{x}\in R} f(\mathbf{x})\, p(\mathbf{x})\, d\mathbf{x} \Big/ \int_{\mathbf{x}\in R} p(\mathbf{x})\, d\mathbf{x} \quad >> \quad \bar{f} \tag{2.3}$$

where $\bar{f}$ is the average over the entire input space (1.7). An important property of any such subregion is its size ("support")

$$\beta_R = \int_{\mathbf{x}\in R} p(\mathbf{x})\, d\mathbf{x}. \tag{2.4}$$

As will be seen there is generally a trade-off between the values of $\bar{f}_R$ and β_R in that larger values of the latter tend to lead to smaller values of the former.

Straightforward estimates of these quantities (2.3) (2.4) respectively will be used:

$$\hat{\beta}_R = \frac{1}{N}\sum_{\mathbf{x}_i\in R} 1(\mathbf{x}_i \in R), \quad \bar{y}_R = \frac{1}{N\cdot\hat{\beta}_R}\sum_{\mathbf{x}_i\in R} y_i, \tag{2.5}$$

where $\{y_i\}_1^N$ are the observed (noisy) output values (1.1) (1.3). Although $\{y_i\}_1^N$ are used to perform the averages it is important to keep in mind that the quantity of interest is $\bar{f}_R$ (2.3). This will have important implications on the strategy employed.

3. Direct applications

There are many circumstances in which attention is directly focused on the extremes of the target function $f(\mathbf{x})$ (1.3). In forecasting future returns of financial securities one is generally interested in identifying those that provide the highest return. Here the output y is the return and the inputs $\mathbf{x}$ might be past returns and various economic factors. The support β_R (2.4) represents the fraction of potential bets. In market research y might represent some customer behavior and the inputs $\mathbf{x}$ would be various demographic variables. The support β_R is the size of the identified market segment. In medical applications the output could be a measure of severity of an illness and the inputs various symptoms and medical measurements. The goal would be to identify the characteristics of the most severely ill patients, perhaps for further diagnostic testing or especially aggressive treatment. Here the support is the fraction of patients for which such testing or treatment is feasible.

In industrial process control one is often interested in maximizing some measure of quality (strength, durability, etc.) of the resulting product. Here y is the quality measure and the inputs $\mathbf{x}$ are various parameters that control the process (concentrations of various chemicals, temperatures, mixing times, etc.). Sometimes there are various characteristic quantities $\{z_k\}_1^K$ associated with a product (operating voltages, currents, impedances) that have associated target values $\{t_k\}_1^K$. The goal is to find values of the process control parameters $\mathbf{x}$ that result in characteristic values simultaneously close to their targets. Here y could be taken to be

$$y = -\sum_{k=1}^{K}(z_k - t_k)^2.$$

In these process control applications the support β_R is generally of no direct interest. It can be viewed as a "meta"-parameter of the procedure whose value is taken to be as small as possible consistent with maximizing $\bar{f}_R$ (2.3) as opposed to $\bar{y}_R$ (2.5).

In all of these applications, as well as most others in data analysis, the output variable y represents noisy measurements of the target $f(\mathbf{x})$. The set of observed inputs seldom completely characterize all possible factors that influence values of the output.

4. Indirect applications

Besides those that directly focus on optimization, there are many other commonly applied data analytic procedures that can be cast within that framework. One such example is classification. Here the actual output quantity assumes (unorderable) categorical values $\{c_k\}_1^K$. In this case there are K ("dummy") output variables $\{y_k = 1(\text{class} = c_k)\}_1^K$. The inputs $\mathbf{x}$ represent the predictor variables. The goal is to identify those regions of the input space within which an observation is most likely to be from one of the individual classes. These are the regions where the corresponding target

$$f_k(\mathbf{x}) = E[y_k \,|\, \mathbf{x}] = \Pr(y_k = 1 \,|\, \mathbf{x})$$

is larger than that of any other class. Thus, classification can be viewed as finding regions in which each $f_k(\mathbf{x})$ is relatively large. The support (2.4)(2.5) of each of the regions can be regarded as meta-parameters jointly optimized to maximize classification accuracy if prediction is the goal, or chosen to aid interpretability in more data analytic situations.

Another commonly applied data analytic procedure is clustering. Here the goal is to find regions of the data space that are relatively densely populated. That is, the data probability density $p(\mathbf{x})$ is large compared to some designated reference density $p_0(\mathbf{x})$ usually (but not always) taken to be a uniform distribution over the range of the data. Regions where the ratio

$$r(\mathbf{x}) = p(\mathbf{x}) \big/ p_0(\mathbf{x}) \tag{4.1}$$

is large represent local concentrations of data in excess of that predicted by $p_0(\mathbf{x})$ ("clusters"). Regions where $r(\mathbf{x})$ is small ($-r(\mathbf{x})$ large) represent local lack of data ("holes") that also may be of interest.

The ratio (4.1) can be maximized by assigning an output value $y = 1$ to each data observation. A Monte Carlo sample of comparable size is then generated from the reference distribution $p_0(\mathbf{x})$ and assigned the output values $y = 0$. The target function

$$f(\mathbf{x}) = E[y \,|\, \mathbf{x}] = r(\mathbf{x}) \big/ (1 + r(\mathbf{x}))$$

(over the pooled data) is monotonic in $r(\mathbf{x})$ so that its maxima/minima represent clusters/holes.

A problem closely related to clustering is association. Here one seeks regions of the data space that are more/less densely populated than would be the case if the data variables were independent of each other. In this case $p_0(\mathbf{x})$ is taken to be the density closest to that of the data $p(\mathbf{x})$ under the constraint of independence

$$p_0(\mathbf{x}) = \prod_{j=1}^{n} p_j(x_j), \tag{4.2}$$

where each factor in the product is the marginal distribution of the respective data variable

$$p_j(x_j) = \int p(\mathbf{x}) \prod_{j' \neq j} dx_{j'}. \tag{4.3}$$

A random sample from this independent density (4.2) (4.3) is easily generated from the data itself. Each x_j-value for each ("Monte Carlo") observation is randomly selected with equal probability from the collection of all x_j data values.

As the above discussion illustrates many data analysis problems can be cast in an optimization framework where the objective function is observed with superimposed noise. Procedures that locate regions of the input space where this target assumes relatively large values represent (in principal) potential solutions to these problems.

5. Interpretability

In descriptive data analysis interpretability becomes an important issue. One would like to restrict solutions to those that can be described and interpreted in terms of important characteristics of the problem, even if this may sacrifice some power. For the problem considered here, this implies that the solution region R (2.3) be specified by simple statements (logical conditions) involving the values of the individual input variables $\{x_j\}_1^n$. Such "rules" take the form

$$R = \bigcup_{k=1}^{K} B_k. \tag{5.1}$$

That is, the solution region is taken to be the union of a set of simply defined subregions $\{B_k\}_1^K$. Let s_{jk} represent a subset of the possible values of input variable x_j; that is $\{s_{jk} \subseteq S_j\}_1^n$ where each S_j (2.1) represents all possible x_j-values. Then each B_k (5.1) is taken to be a "box"

$$B_k = s_{1k} \times s_{2k} \times \cdots \times s_{nk} \tag{5.2}$$

within the entire input space (2.2). Each box (5.2) is described by the intersection of subsets of values of each input

$$\mathbf{x} \in B_k = \bigcap_{j=1}^{n} (x_j \in s_{jk}). \tag{5.3}$$

For inputs that are real (perhaps discretely) valued, the subsets are represented by contiguous subintervals

$$s_{jk} = [t_{jk}^-, t_{jk}^+]. \tag{5.4}$$

Thus, the projection of a box B_k on the subspace of real valued inputs is a hyper-rectangle. For categorical inputs the individual subsets of values s_{jk} are explicitly delineated. Note that for the case in which the subset of values (real or categorical) is in fact the entire set $s_{jk} = S_j$, the corresponding factor $x_j \in S_j$ can be omitted from the box definition (5.3). In this case it takes the simpler form

$$\mathbf{x} \in B_k = \bigcap_{s_{jk} \neq S_j} (x_j \in s_{jk}). \tag{5.5}$$

The particular input variables x_j for which $s_{jk} \neq S_j$ are said to be those that "define" the box B_k. As an example, for the marketing data base described in Section 15 the rule

$$\mathbf{x} \in B_k = \begin{cases} 18 < \text{age} < 34 \quad \& \\ \text{marital status} \in \{\text{single, living together-not married}\} \quad \& \\ \text{householder status} = \text{rent} \end{cases}$$

defines one of the boxes comprising the subregion associated with high frequency of visiting bars and night clubs.

6. Covering

From (2.3) (5.1) one sees that the goal of the optimization procedure is to induce a set of boxes (5.5) from the data (1.1) that collectively cover the region of the input space where the target $f(\mathbf{x})$ assumes large values. Given an algorithm for constructing boxes from data (see Section 7) this can be accomplished by "covering" [see Mitchell(1997)]. The same box construction algorithm is repeatedly applied in a sequential manner to subsets of the data. The first box B_1 is induced using the entire data set (1.1). The second B_2 is constructed using the original data with those observations covered by the first box removed: $B_2 \sim \{y_i, \mathbf{x}_i \,|\, \mathbf{x}_i \notin B_1\}$. At the Kth iteration a box B_K is induced using the data remaining after the removal of all observations covered by the $K-1$ previously induced boxes: $B_K \sim \{y_i, \mathbf{x}_i \,|\, \mathbf{x}_i \notin \bigcup_{k=1}^{K-1} B_k\}$. This continues until either the estimated target means within the boxes

$$\bar{y}_K = ave[y_i \,|\, \mathbf{x}_i \in B_K \;\&\; \mathbf{x}_i \notin \bigcup_{k=1}^{K-1} B_k] \tag{6.1}$$

become too small, say less than the global mean $\bar{y}$ (1.8), or their individual support

$$\beta_K = \frac{1}{N} \sum_{i=1}^{N} 1(\mathbf{x}_i \in B_K \;\&\; \mathbf{x}_i \notin \bigcup_{k=1}^{K-1} B_k) \tag{6.2}$$

becomes too small.

The set of boxes induced in this manner can then be used to form the final region (5.1) according to the specific data analytic goal. One might select those whose mean (6.1) is greater than some threshold $\bar{y}_0$

$$R = \bigcup_{\bar{y}_k > \bar{y}_0} B_k,$$

or perhaps choose the subset that yields the largest region mean $\bar{y}_R$ (2.5) for a specified support β_t

$$\beta_R = \sum_{k=1}^{K} \beta_k \simeq \beta_t.$$

Here $\{\beta_k\}$ are the individual box supports (6.2). Alternatively, one can regard the sequence of induced boxes, along with their respective means, as an ordered set ("decision list") [Rivest (1987)]

$$\{\bar{y}_k, B_k\}_1^K. \tag{6.3}$$

An input point $\mathbf{x}$ that is covered by more than one box is assigned to the one appearing first in the list, and associated with its box mean.

7. Box induction

Central to the optimization algorithm is the procedure used to construct the individual boxes. Given the data (or a subset thereof), its goal is to produce a box B within which the target mean

$$\bar{f}_B = \int_{\mathbf{x} \in B} f(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} \,\big/ \int_{\mathbf{x} \in B} p(\mathbf{x}) d\mathbf{x} \tag{7.1}$$

is as large as possible consistent with the interpretability constraint (5.5). The strategy employed is one of "patient" top-down successive refinement followed by bottom–up recursive expansion.

7.1. Top–down peeling

This first phase of the box induction strategy begins with a box B that covers all of the data. At each step (iteration) a small subbox b within the current box B is removed. The particular subbox b^* chosen for removal, to produce the next (smaller) box in the sequence, is the one that yields the largest output mean value within the next box $B - b^*$

$$b^* = \arg\max_{b \in C(b)} ave[y_i \,|\, \mathbf{x}_i \in B - b]. \tag{7.2}$$

Here $C(b)$ represents a class of potential subboxes eligible for removal. The current box is then updated

$$B \leftarrow B - b^* \tag{7.3}$$

and the procedure repeated on this newer smaller box. This "peeling" away of small subboxes continues until the support within the current box β_B falls below some threshold value β_0

$$\beta_B = \frac{1}{N}\sum_{i=1}^{N} 1(\mathbf{x}_i \in B) \leq \beta_0. \tag{7.4}$$

The quantity β_0 is a "meta"-parameter of the procedure. A choice for its value involves both statistical and problem domain dependent considerations. These are discussed in Section 9.

The class of boxes $C(b)$ eligible for peeling (7.2) is dictated by the interpretability constraint (5.5). Each eligible subbox b is defined by a single input variable x_j.

(a) Real valued inputs each provide two eligible subboxes, b_{j-} and b_{j+}, which border the respective lower and upper boundaries of the current box B on the jth (real) input

$$\begin{aligned} b_{j-} &= \{\mathbf{x} \,|\, x_j < x_{j(\alpha)}\} \\ b_{j+} &= \{\mathbf{x} \,|\, x_j > x_{j(1-\alpha)}\}. \end{aligned} \tag{7.5}$$

Here $x_{j(\alpha)}$ is the α-quantile of the x_j-values for data within the current box, and $x_{j(1-\alpha)}$ is the corresponding $(1-\alpha)$-quantile. The quantity α is another meta-parameter usually taken to be quite small ($\alpha \leq 0.1$). Particular values are dictated by statistical considerations detailed in Section 8.

(b) Each categorical variable x_j contributes a set of eligible boxes, one for each of its values s_{jm} within the current box

$$b_{jm} = \{\mathbf{x} \,|\, x_j = s_{jm}\}, \quad s_{jm} \in S_j. \tag{7.6}$$

The complete class $C(b)$ of eligible subboxes (7.2) is the collection of all of those defined on the respective input variables. The data within each one is (in turn) provisionally removed and the mean over the remaining observations in the current box calculated. The data within the subbox b^* that gives rise to the largest such (remaining) mean is then permanently removed to define the next smaller box (7.3).

The use of various types of peeling procedures in the context of multivariate statistical analysis was introduced by Barnett (1976) and employed by Green (1981). Donoho and Gasko (1992) used this concept to define robust affine equivariant estimators of multivariate location.

7.2. Bottom–up pasting

The goal of the top–down peeling algorithm is a box covering a subregion of the input variable space within which the target mean (7.1) is relatively large. The boundaries (5.5) of this box are determined by particular values of those variables that defined the subboxes (7.5) (7.6) chosen for peeling (7.2) at the various stages of the top–down refinement procedure. Except for the last one, these final box boundaries were determined at earlier steps of the peeling sequence without knowledge of later peels that further refined the boundaries on other input variables. It is therefore possible that the final box can be improved by readjusting some of its boundaries. This is done through a bottom–up "pasting" strategy.

This pasting algorithm is basically the inverse of the peeling procedure. Starting with the peeling solution, the current box B is iteratively *enlarged* by pasting onto it a small subbox b^*, $B \leftarrow B \bigcup b^*$. The small subbox b^* chosen is the one from an eligible class $b \in C(b)$ that maximizes the output mean in the new (larger) box. The eligible class of pasting subboxes is defined analogously to those used for peeling (7.5) (7.6). For each input variable x_j, eligible subboxes b are bounded by the current box boundaries on all other variables $\{x_{j'}\}_{j' \neq j}$. For categorical variables x_j, values s_{jm} (7.6) not represented in the current box B define subboxes b eligible for pasting. For real valued x_j, eligible subboxes are represented by small intervals extending the upper and lower boundaries of B on that variable. The widths of each of these intervals is chosen so as to contain αN_B observations, where α is the peeling fraction (7.5) and N_B is the number of observations in the current box B.

Bottom-up pasting is iteratively applied, successively enlarging the current box, until the addition of the next subbox b^*causes the output mean $\bar{y}_{B+b^*}$ to decrease. At that point pasting stops and the current box B becomes the solution. Although bottom–up pasting usually provides some improvement, it seldom has a dramatic effect. There are occasions however when it does produce substantial improvement to the peeling solution.

8. Patient rule induction

There are two principal meta-parameters that control box induction; they are the peeling fraction α (7.5) and the support β_0 (7.4) of the peeling solution. Statistical performance considerations govern the choice of a value for α. These are discussed in this section. Choosing a value for β_0 involves both statistical and application domain dependent considerations. These are discussed in Section 9.

In order to discuss statistical performance a formal (idealized) goal must be specified as well as a measure of how well that goal is achieved. For a given value of β_0 the formal goal of box induction can be expressed as maximizing the mean of the target function $f(\mathbf{x})$ within a box (7.1) with respect to the box parameters s_{jk} (5.3), under a constraint on box support, $\beta = \beta_0$. That is,

$$\bar{f}^* = \max_B ave[f(\mathbf{x}) \,|\, \mathbf{x} \in B]; \quad \beta_B = \beta_0. \tag{8.1}$$

Let $\hat{f}$ be the solution (estimate) induced by the peeling algorithm when applied to data. Then one measure of performance is average (expected) mean-squared error

$$E[\bar{f}^* - \hat{f}]^2 = (\bar{f}^* - E\hat{f})^2 + E[\hat{f} - E\hat{f}]^2. \tag{8.2}$$

The expected value (8.2) represents the squared-error of $\hat{f}$ averaged over all data sets of size N that could have been realized from the system under study. The actual data at hand (1.1) is presumed to be one of these data sets drawn at random. The quantity $E\hat{f}$ is the average over all data sets of the estimated solution for each one. The first term on the right of (8.2) is the deviation of this average from the truth and represents the squared systematic error ("bias-squared"). The second right hand term is the variance of the solutions over all data sets. It characterizes the instability of the procedure. High instability implies that the solution is very sensitive to small changes in the data. This not only reduces accuracy, but it is especially troublesome from the perspective of interpretation. It is dangerous to place high confidence on induced features (i.e. box boundaries) if slight changes to the data strongly affect the nature and/or existence of those features.

8.1. Fragmentation

Top-down successive refinement procedures such as peeling can be viewed as steepest ascent with a limited number of steps. Each iteration produces the step that is estimated to provide the greatest local increase in the objective function, here box mean. This "greedy" step is seldom optimal in terms of being the one that moves closest to the ultimate solution. If a very large number of steps is possible then this lack of optimality represents only a computational issue; a large enough number of greedy steps will eventually arrive at a maximum.

In the case of top-down successive refinement, a large number of steps is not possible. This is due to data fragmentation. Each step reduces the amount of data available to the next step. At some point there is not enough remaining data to continue. The box support β has fallen below the stopping threshold β_0.

For a given data set size N the number of steps is determined by the amount of fragmentation ("greed") permitted at each one. With a greedy strategy a large fraction of the data is removed at each step allowing for only a small number of iterations. This restricts the ability of the procedure in later steps to compensate for early steps that may have been highly suboptimal. This can have detrimental effect on both bias and variance (8.2) leading to increased mean squared error. Bias is induced when the best step at an early iteration is unambiguous but, due to the nature of the target function, is misdirected. Variance is induced when the best step is ambiguous; there are several possibilities that provide similar local improvement in target mean. In this case the choice among them is driven largely by the noise, making it highly sensitive to small changes in the data. With a greedy strategy there are only a few steps so each one has a large effect on the final outcome. There is little opportunity to mitigate the damage caused by a bad step.

In the context of box induction very greedy strategies pose another potential problem. Only a small number of input variables can be involved in defining the box boundaries (5.5). In order for an input to participate it must be chosen for peeling at least once in the top-down sequence. If the number of inputs n is large and the number of steps small, only a small fraction of the input variables can be involved. While this may represent an interpretational advantage, it can introduce severe bias and variance when more than a few inputs are required to isolate the target maximum.

8.2. Patience

The problem of fragmentation with top-down refinement procedures can be mitigated by adopting a "patient" strategy; only a small fraction of the data in the current box is removed at each iteration. This makes each individual step less important to the final outcome (variance) and permits greater opportunity for later steps to take advantage of structure uncovered by earlier ones and/or to compensate for those that were misdirected (bias). Also, more input variables are permitted (but not required) to enter into the definition of the induced box.

For real valued input variables, the degree of patience of the box induction algorithm (Section 7.1) is controlled by the value of the peeling parameter α (7.5). If all inputs (chosen for peeling) are real with distinct values the number of steps (peels) L is

$$L = \log \beta_0 \big/ \log(1 - \alpha).$$

The above discussion suggests choosing the smallest value possible for α in order to achieve the maximum degree of patience. This would be $\alpha = 1/N_B$ where N_B is the number of observations in the current box; one observation is peeled off at each step.

However the parameter α serves another role; it is also a smoothing parameter. The goal of each peel is to maximize the average of the target function $f(\mathbf{x})$ in the next smaller box $B - b$. The variance of the estimate of this is proportional to

$$\frac{1}{\alpha} var[\varepsilon \mid \mathbf{x} \in b] \tag{8.3}$$

where ε is the random noise (1.3). Since the goal is to have the peeling driven by signal $f(\mathbf{x})$ rather than the noise ε, the value of α cannot be made too small. There is a trade–off between the goals of patience and accurate peeling that depends on the signal to noise ratio and total sample size. Empirical evidence so far suggests that the peeling procedure is not sensitive to the value of α provided it is not too large; patience appears to be the more important goal. Values in the range $0.05 \leq \alpha \leq 0.1$ seem to work well. Methods for estimating an optimal value from the data at hand are discussed in Section 13.

For input variables that are categorically valued, or real with a small number of distinct values, the degree of patience cannot be controlled with precision. All observations with identical values on an

input variable are considered together. Patience is encouraged by making only a single value eligible for peeling at each step. Thus, peeling on variables with more (discrete) values induces more patience. Methods for encouraging patience in this context are discussed in Section 14. Owing to its emphasis on patience, the procedure described herein is referred to as a patient rule induction method ("PRIM").

9. Stopping rule

Peeling stops when the support β_B of the current box B is below a chosen threshold β_0 (7.4). Choice of a value for β_0 depends on the data analytic goal. Sometimes the goal is a "point" estimate of the location of the function maximum. The industrial process control problems discussed in Section 3 are examples. In this case one would like the value of β_0 to be as small as possible consistent with accurate estimation of a maximum of $f(\mathbf{x})$. Because the function is measured with noise (1.3) it can be counter productive to allow the value of β_B to become arbitrarily small. The peeling procedure uses the output mean $\bar{y}_B$ (7.2) as an estimate of the function mean $\bar{f}_B$ (7.1) in each successive box. This estimate becomes less reliable with smaller values of box support β_B and the peeling procedure can be distracted by noise; the value of $\bar{y}_B$ can continue to increase whereas the value of $\bar{f}_B$ actually decreases. This "over-fitting" phenomenon is common to all procedures that optimize on random data.

9.1. Cross–validation

A common procedure used to deal with over-fitting is cross-validation. The data set is randomly partitioned into two parts, a "learning" data set and a "test" data set. Typically the learning set is taken to be twice the size of the test set. The peeling procedure is applied to the learning data with a very small value for β_0, allowing for very small box supports. The test data set is then used to estimate the output mean $\bar{y}_B$ in each successive box of the peeling sequence induced on the training data. If the noise associated with the test data is independent of that for the training data this will produce unbiased estimates of the target mean $\bar{f}_B$ in each box. The box with the largest associated (test) mean is taken as the estimated optimal one containing the target maximum.

9.2. Application domain

Often, application domain considerations place constraints on the support of the induced box. The support of the cross-validated estimate of the optimal box may be too small to be useful. It may identify characteristics of financial securities with very high return, but the support may be too small to allow sufficiently frequent betting. A highly desirable market segment may be located but be too small to be worth targeting. In some applications there may be a profitability threshold and one seeks boxes with largest support whose mean is above that threshold. In all of these cases there may be boxes that have much larger support than the estimated optimal one, but only slightly smaller estimated target mean. These "suboptimal" boxes may be more useful in particular applications.

Application domain judgements such as the trade–off between box mean and support are best made by the user. The user can be presented with the mean and support of each of the L boxes in the (nested) peeling sequence $\{B_l\}_1^L$,

$$\{\bar{y}_l, \beta_l\}_1^L, \tag{9.1}$$

as computed on the (left out) test data set. The one most appropriate for the application at hand can then be chosen. This may be the "optimal" one with largest estimated mean or one that has a smaller mean but larger support. After this choice is made, bottom–up pasting (Section 7.2) can then be applied to further improve the mean–support trade–off.

Figure 9.1 shows a scatter plot of $\bar{y}_l$ against β_l (9.1) for the first sequence of boxes induced from the geology data described in Section 11. Such a plot will be referred to as a "peeling trajectory". The box mean $\bar{y}_l$ is seen to steadily increase with decreasing support β_l with the exception of three small bumps. The optimal solution is $\bar{y}_{35} = 0.97$ at $\beta_{35} = 0.028$. If the support of this box is too small another candidate might be $\bar{y}_{24} = 0.95$ at $\beta_{24} = 0.081$. It has slightly lower estimated mean but almost three times the support. Other choices may represent more useful alternatives and the user

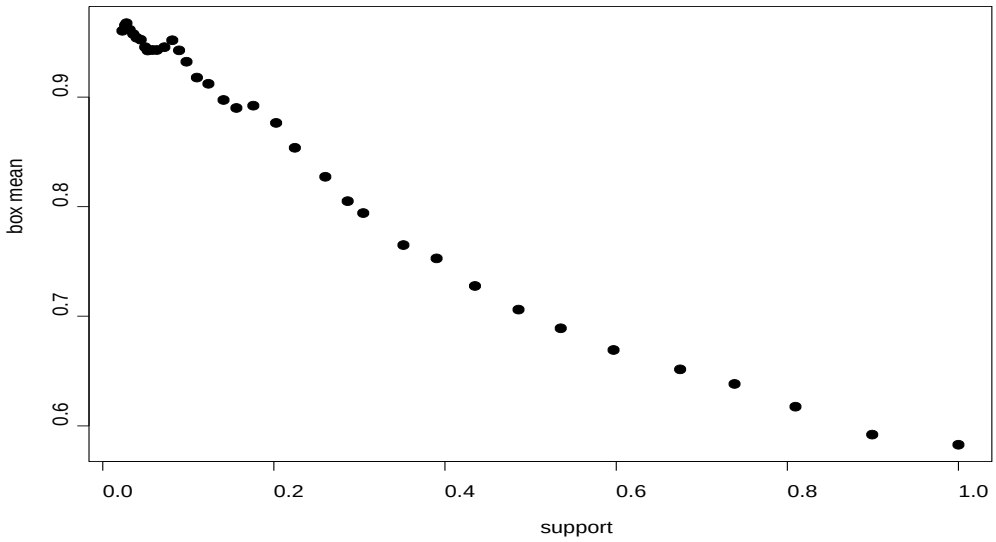

Figure 9.1: First peeling trajectory for the geology data. Mean vs. support for boxes induced by top–down peeling. Most appropriate trade–off is chosen by the user.

is free to choose the one most appropriate. Involving the user in this manner helps insure that the induced boxes best serve the particular data analytic goal.

10. Redundant input variables

In addition to target mean and support, a box B is characterized by the set of input variables that define it (5.5). The "complexity" of a box can be defined as the cardinality of this set. For a given mean and support, less complex boxes are preferred due to their simpler interpretation. As noted in Section 8.2 an important property of patient rule induction is that it allows more complex boxes to be induced. This represents an important advantage when such complexity is required, reducing bias. However, it has the side effect that nonessential inputs can also enter into the box definition. These "redundant" inputs cause increased complexity, thereby clouding interpretation without reducing bias.

There are two mechanisms that can cause redundant inputs, noise and collinearity. The noise ε (1.3) induces variance in the estimate of the best peeling variable at each iteration. This can cause irrelevant variables to be selected owing to sampling fluctuations. As discussed in Section 8.2 this is the motivation for not choosing the peeling parameter α too small (8.3). Although effective, this strategy cannot perfectly screen out irrelevant input variables.

Collinearity occurs when there is a high degree of correlation among two or more input variables within the solution box. In this case, only one of the variables in the correlated group is needed as part of the box definition, namely the one with the most restrictive range of values. By virtue of its correlations with the others in the group, restricting its range alone serves to restrict the ranges of the other variables.

Whether induced by noise or collinearity, redundant input variables can be profitably deleted from the set (5.5) defining the box. In the case of irrelevant inputs induced by noise this can serve to increase accuracy in terms of providing higher box mean and support. Even when this is not the case the interpretative value associated with removing marginally relevant inputs can sometimes outweigh a small decrease in estimated box mean. Such decisions are likely to depend on problem specifics and,

as with choosing box support, are best judged by the user. Domain knowledge can sometimes help in deciding whether particular input variables that appear marginal on statistical grounds actually are likely to be relevant or not.

In order to make such judgements a statistical measure is required of each input variable's relevance in defining the box. One such measure is the decrease in box mean when the variable is removed. Let x_j be one of the variables involved in the box definition (5.5). It can be removed by replacing its corresponding subset of values s_{jk} by the set of all possible x_j-values S_j,

$$s_{jk} \leftarrow S_j. \tag{10.1}$$

The corresponding decrease in box mean is then recorded. The defining input with the smallest such decrease is deemed to be the least relevant one and is (provisionally) removed. This procedure is then repeated on the remaining inputs identifying the second least relevant input variable and it is provisionally deleted. This continues until there is only one input variable left in the box definition. This one is regarded as the most relevant box variable. The sequence of deleted inputs induced in this manner orders the input variables in ascending relevance to box the definition.

Table 10.1
Sequential variable relevance for the geology data.

var.	$\bar{y}$	β
0	0.946	0.072
1	0.952	0.081
6	0.941	0.108
2	0.755	0.340
5	0.583	1.0

Table 10.1 presents the results of this backward stepwise elimination procedure on the box ($\bar{y} = 0.95$, $\beta = 0.07$) of the geology data peeling trajectory shown in Fig. 9.1. It is seen to be defined by four (out of 11) input variables $j \in \{1, 2, 5, 6\}$. The first row of Table 10.1 represents this box and the next four rows represent the results of removing each defining variable (in turn) in order of estimated relevance. The first column labels the variable removed and the next two are respectively the box mean and support as a result of the removal, estimated from the (left out) test data.

One sees from Table 10.1 that removing x_1 from the box definition actually provides a slight *increase* in estimated box mean. Removing x_1 and x_6 results in a slight decrease whereas removing x_1, x_6, and x_2 produces a dramatic decrease of target mean inside the box now defined only by x_5. Of course, also removing x_5 produces the starting box enclosing all of the data. Note that box support always increases as variables are removed.

Given a box chosen from the peeling trajectory (Fig. 9.1) the corresponding results from the backward stepwise procedure, as represented in Table 10.1, can be employed by the user to judge trade-offs between box complexity, mean, and support. In this way the user's domain knowledge as well as data analytic requirements can be taken into account. For this particular example (Table 10.1) the choice is rather obvious. Removing x_1 and x_6 produces a box with one half the complexity and 50% more support than the originally induced one, with only slightly lower estimated target mean. In many other situations, however, the choices are not so obvious and the user's knowledge and judgement become valuable assets.

11. Geology data

The data set used for illustration in the preceding two sections consists of a sample of $N = 13317$ garnets collected from around the world [Griffin et. al. (1997)]. A garnet is a complex Ca-Mg-Fe-Cr silicate that commonly occurs as a minor phase in rocks making up the Earth's mantle. The input variables are the concentrations of various chemicals measured on each garnet

$$\mathbf{x} = \{TiO_2,\ Cr_2O_3,\ FeO,\ MnO,\ MgO,\ CaO,\ Zn,\ Ga,\ Sr,\ Y,\ Zr\}.$$

One data analytic objective was to contrast the chemical compositions of garnets with different plate–tectonic settings. Three such plate–settings are represented in the data: (1) ancient stable shields, (2) Proterozoic shield areas, and (3) young orogenic belts. The output variable chosen for illustration was an indicator of being from the first setting

$$y = 1(\text{plate-tectonic setting } = \text{ancient stable shields}).$$

The global output mean is $\bar{y} = 0.59$; 59% of the garnets in the sample had this setting.

The first box induced by the PRIM procedure using the choices described in the previous sections is

$$\mathbf{x} \in B_1 = \text{Cr}_2\text{O}_3 \geq 0.62 \quad \& \quad \text{MgO} \geq 0.67. \tag{11.1}$$

Here the limits are stated in terms of the quantiles of the corresponding individual concentrations over the sample. Thus this box covers garnet samples for which the concentration of Cr_2O_3 is among its 38% highest values and that for MgO assumes values in its top 33%. The output mean in this box is $\bar{y}_1 = 0.94$ with support $\beta_1 = 0.11$. That is, this box covers 11% of the data and 94% of the garnets within it were ancient stable shields. The next two boxes induced through the covering procedure outlined in Section 6 are

$$\begin{aligned} \mathbf{x} &\in B_2 = \text{Ga} \geq 0.86 \quad \& \quad 0.09 \leq \text{Y} \leq 0.92, \\ \mathbf{x} &\in B_3 = \text{Cr}_2\text{O}_3 \geq 0.43 \quad \& \quad \text{MgO} \geq 0.69 \quad \& \quad \text{CaO} \leq 0.47 \end{aligned} \tag{11.2}$$

with respective output means $\bar{y}_2 = 0.80$ and $\bar{y}_3 = 0.83$. The respective supports are $\beta_2 = 0.10$ and $\beta_3 = 0.09$. The first three boxes $R = B_1 \bigcup B_2 \bigcup B_3$ collectively cover 29% of the data ($\beta_R = 0.29$) with target mean $\bar{y}_R = 0.86$. In the entire sample the relative odds of a garnet having the ancient stable shields setting is 1.44 whereas in the induced region R the corresponding odds are 6.14. Thus, the odds ratio has been increased by over a factor of four. The three rules (11.1) (11.2) provide a parsimonious and easily interpreted description of the induced region.

In the geology data base there are 493 garnets with the third plate–tectonic setting, comprising only 3.7% of the entire sample. Thus, the output

$$y = 1(\text{plate-tectonic setting } = \text{young orogenic belts}) \tag{11.3}$$

can be viewed as a "rare event" phenomenon, $\bar{y}_3 = 0.037$. Applying the PRIM procedure to (11.3) produced the (first) box

$$\mathbf{x} \in B_1 = \left\{ \begin{array}{l} \text{Cr}_2\text{O}_3 \leq 0.16 \quad \& \quad \text{MgO} \geq 0.42 \quad \& \\ \text{CaO} \geq 0.29 \quad \& \quad \text{Ga} \leq 0.59 \quad \& \\ 0.73 \leq \text{Y} \leq 0.96. \end{array} \right. \tag{11.4}$$

Its mean value is $\bar{y}_1 = 0.82$ with support $\beta_1 = 0.017$. Thus, almost 40% of the young orogenic belts garnets are contained in this very small box comprising only 1.7% of the entire sample. The relative odds of finding this type of garnet in the box are increased by over a factor of 100 from that of the entire data base. The next two induced boxes (not shown) have mean and support values ($\bar{y}_2 = 0.12$, $\beta_2 = 0.04$) and ($\bar{y}_3 = 0.19$, $\beta_3 = 0.025$) with corresponding odds ratio increases of 3.55 and 6.11 respectively. These results suggest that a substantial fraction of the young orogenic belts garnets have chemical compositions very distinct from those of the other two settings. The induced rules (e.g. (11.4)) provide insight as to the nature of these distinctions.

12. Missing values

In the geology data base there were no missing values; all concentration measurements were present for each garnet. However in other contexts such as marketing data, sample survey questionnaires, and even in some scientific data sets, missing values are a problem. In such settings a strategy is needed to enable the procedure to effectively process the available data and not be distracted by values that

are not present. Such a strategy should accomplish two goals. First, if there is an association between the output variable and the probabilities of missing values on various inputs, this should be used to advantage. Second, even in the absence of such associations, one would like the procedure to effectively use values of input variables highly correlated with relevant ones as surrogates when the values of the corresponding relevant variables are missing.

In the context of patient rule induction both goals can be met with a surprisingly simple strategy; the value $x_j =$ missing is simply taken as another (legitimate) value that it can assume. For categorical variables this represents no change to the peeling/pasting algorithms (Section 7). For each real valued variable x_j the number of peeling subboxes b (7.5) that it contributes to the eligible class $C(b)$ (7.2) is extended to three

$$\begin{aligned} b_{j-} &= \{\mathbf{x} \,|\, x_j < x_{j(\alpha)}\} \\ b_{j+} &= \{\mathbf{x} \,|\, x_j > x_{j(1-\alpha)}\} \\ b_{j0} &= \{\mathbf{x} \,|\, x_j = \text{ missing}\}. \end{aligned} \tag{12.1}$$

The observations in the current box B for which $x_j =$ missing are of course not used to compute $x_{j(\alpha)}$ and $x_{j(1-\alpha)}$.

This strategy directly meets the first goal. Associations between the output and missing input values are clearly incorporated. It also *indirectly* meets the second goal. The mean y-value for those observations missing a particular input value ($x_j =$ missing), through a mechanism unrelated to the value of the output, is likely to be close to the global target mean $\bar{y}$. As such, the missing value is unlikely to be chosen for peeling, especially in the early stages of the top-down sequence. If x_j is a highly relevant input variable, other inputs strongly correlated with it appear relevant by virtue of that correlation. Since the original highly relevant variable is unlikely to choose its missing value for peeling, those variables correlated with it that contain values when those of the original one are missing have increased likelihood of being chosen for peeling.

This missing value strategy has the (perhaps) unpleasant side effect of clouding interpretation. When (fundamentally) irrelevant input variables are chosen for peeling because of their strong correlations with relevant ones they are important to the final box definition only because of the missing values on the relevant ones. They serve only as surrogates and have no direct impact on the output variable. In the absence of missing values they would have been much less likely to be chosen for peeling and, if chosen, much more likely to be removed through the redundant variable strategy outlined in Section 10.

13. Multiple trajectories

A peeling trajectory (9.1) (Fig. 9.1) is induced by an application of the peeling algorithm. It allows the user the opportunity to choose the box mean–support trade–off according to judgement and programmatic needs. The particular trajectory induced depends upon the data and the details of the algorithm. These details involve values of meta-parameters that control various aspects of the procedure. One such is the peeling parameter α (7.5). Others are discussed in Section 14. Although the procedure is fairly stable, different parameter values can induce (at least slightly) different trajectories and the ("model selection") issue arises of which one is best from the perspective of the user's needs. As with other aspects of the procedure such judgements are best made by the user.

The model selection strategy adopted here is to simultaneously present a large number of trajectories on a single plot, each based on its particular set of meta-parameter values. Each trajectory produces a set of points like those of Fig. 9.1 representing the mean–support trade–offs for its nested sequence of boxes. The union of all such trajectories produces a scatter plot where each point represents a box from one of the individual trajectories. As with a single trajectory (Fig. 9.1) the user chooses the box represented by the point on the scatter plot corresponding to the preferred mean–support trade–off. The issue of redundant input variables for that box can then be addressed in the manner described in Section 10 (e.g. Table 10.1). In practice several such boxes are identified representing potentially promising mean–support trade–offs. Each one is then followed by bottom–up pasting and

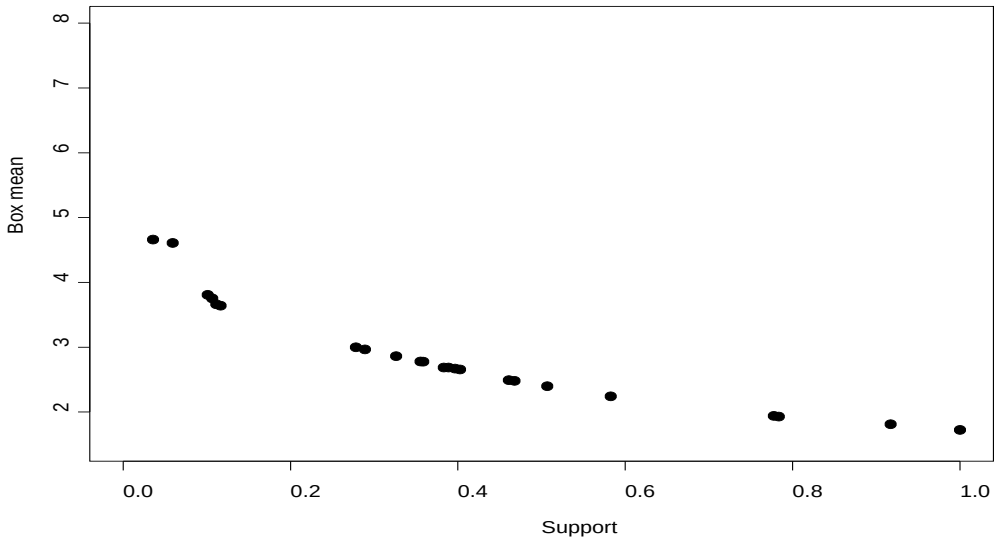

Figure 13.1: Single peeling trajectory on marketing data after thinning, to be contrasted with the corresponding multiple trajectory plot in the next figure.

then redundant variable elimination. The overall best solution, as judged by the user, is then adopted to produce the preferred box at this stage of the covering sequence (Section 6).

Additional trajectories can also be induced by "bumping" [Tibshirani and Knight (1995)]. Here one repeatedly modifies the data supplied to the algorithm rather than modifying the algorithm itself. The purpose is to mitigate instability by producing a large number of randomly generated solutions, each one induced by randomly perturbing the learning data, usually through "bootstrapping" [Efron and Tibshirani (1995)]. The hope is that one of these random solutions may (accidentally) turn out to be better (on the left-out test data set) than the one induced on the original learning data. Although intended for greedy procedures such as decision trees that are inherently very unstable [Breiman (1996)], bumping may also sometimes help with peeling in anomalous situations where there is a high degree of inherent ambiguity among the initial set of peels. In such cases bumping will tend to produce a variety of different initial peeling sequences one or more of which may lead to a better final set of boxes than the sequence derived from the original learning data set.

Whether induced by modifying parameter values or bumping, a large number of trajectories presented on a single plot can produce a confusing picture. The large number of points can obscure details making it difficult to choose a good mean-support trade-off. The plot can be greatly simplified by "thinning" ; the vast majority of points that represent unlikely choices are removed. Any point i that represents a box with smaller mean *and* smaller support than that of another box j

$$\bar{y}_i < \bar{y}_j \quad \& \quad \beta_i < \beta_j \tag{13.1}$$

is not likely to be selected; one would favor the box represented by point j. Therefore all points that are dominated (13.1) by another point on the plot can be removed leaving only the upper envelope of undominated points. This presents a less confusing picture making it easier for the user to make an appropriate choice.

Figure 13.1 shows the single trajectory produced by peeling using criterion (7.2) on the marketing data base described in Section 15 (y = number of round trip flights/year). Figure 13.2 shows a corresponding plot resulting from 20 trajectories induced by 10 bumps on each of two different peeling

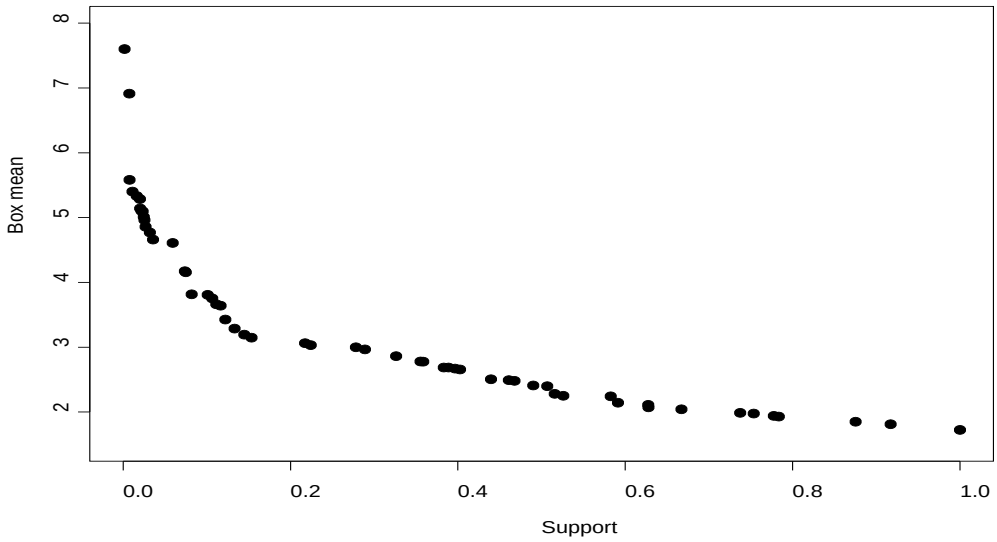

Figure 13.2: Multiple trajectory plot on marketing data after thinning. Compared to the single trajectory plot, it offers more mean–support choices and better boxes at smaller support values.

criteria (Section 14). Both plots have been thinned (13.1). Here, for $\beta \gtrsim 0.05$ the two sets of points track each other fairly closely, with Fig. 13.2 offering a richer set of alternatives from which to choose. For $\beta \lesssim 0.05$ the multiple trajectory plot provides a series of boxes with much higher estimated output mean values. This represents a clear advantage if boxes of small support ($\beta \lesssim 0.05$) are useful in this application. Usually the gains produced by employing multiple trajectories are generally more modest than that observed here. Patient peeling tends to be fairly stable. However, for moderately sized data sets ($N \lesssim 20000$) where computation is not a burden, multiple trajectories can provide insurance against a poor choice of meta-parameter values or, through bumping, against an unfortunate set of (ambiguous) peels.

14. Peeling criteria

The guiding principle of the peeling strategy (Section 7.1) is patience (Section 8). For real valued input variables with many different values, the degree of patience is effectively controlled by the meta-parameter α (7.5). For real valued inputs with fewer distinct values (many ties), subboxes b eligible for peeling are taken to have support β_b as close as possible to the value of α. Tied values are always peeled together. For categorical variables the support β_b of eligible subboxes (7.6) cannot be directly controlled. In these latter two cases it is still desirable to encourage patience to the extent possible.

14.1. Subbox criteria

The procedure discussed in Section 7.1 uses the target mean $\bar{y}_{B-b}$ in the next smaller box $B-b$ (7.2) as the peeling criterion to be maximized with respect to the subbox b. This is equivalent to maximizing the improvement

$$I(b) = \bar{y}_{B-b} - \bar{y}_B \tag{14.1}$$

of that over the current box B. When there are several competing subboxes with similar values of $I(b)$, the one with the smallest support β_b is most desirable since it induces the most patience; more

data remains for further peeling. Therefore, improvement per unit of removed support β_b becomes an issue.

Patience can be encouraged by using a modified peeling criterion

$$J(b) = I(b) \cdot P(\beta_b) \tag{14.2}$$

where $P(\beta_b)$ is a monotone decreasing function of its argument. Using

$$P(\beta_b) = 1/\beta_b \tag{14.3}$$

produces a criterion (14.2) that measures improvement (14.1) per amount of removed support β_b. Expressing (14.1) as

$$I(b) = \frac{\beta_b}{\beta_B - \beta_b} \left[\bar{y}_B - \bar{y}_b\right], \tag{14.4}$$

where $\bar{y}_b$ is the output mean in the peeled subbox b and β_B is the support of the parent B, yields the criterion

$$J(b) = \frac{1}{\beta_B - \beta_b} \left[\bar{y}_B - \bar{y}_b\right] = (\bar{y}_{B-b} - \bar{y}_b)/\beta_B. \tag{14.5}$$

Thus, using (14.3) in (14.2) produces as a criterion the difference between the output mean of the data remaining and that of the peeled subbox. Even more patience can be induced by choosing

$$P(\beta_b) = (\beta_B - \beta_b)/\beta_b \tag{14.6}$$

which produces the mean in the peeled subbox $\bar{y}_b$ as an (equivalent) criterion to be *minimized*.

Each of the alternatives (14.1) (14.3) (14.6) can produce (at least somewhat) different peeling trajectories. All of them can be included together as part of the multiple trajectory strategy described in Section 13. In this way the best one for the application at hand can be chosen by the user.

14.2. Input variable criteria

All of the peeling criteria discussed so far (14.1) (14.3) (14.6) still have a greedy component: the subbox b^* that optimizes them is chosen to produce the next box $B - b^*$. Removing a different subbox at a particular step may in fact produce better boxes later in the peeling sequence. The goal of a patient strategy is to mitigate this effect by providing a large number of peels in the hope that later steps can compensate for poor (greedy) choices. Bumping can also help in this regard. These can be viewed as "passive" strategies. Occasionally, in unusual situations, the use of more "proactive" strategies to overcome greed can produce better trajectories. These can be included among the multiple trajectories from which the user can choose.

The above criteria all focus on subboxes b. The input variables $\{x_j\}_1^n$ serve only to define those eligible for deletion (7.5) (7.6). A more proactive strategy could focus on the input variables themselves, attempting to ascertain which ones at each step are most likely to be influencing the target function $f(\mathbf{x})$ within the subregion defined by the current box, $\mathbf{x} \in B$. One such criterion that requires no additional computation is

$$J_j = \max_m\{J(b_{jm})\} - \min_m\{J(b_{jm})\} \tag{14.7}$$

where $J(b)$ is the original criterion (14.2) and $\{b_{jm}\}$ are the subboxes contributed by jth input variable to the eligible set. The subbox chosen for peeling $b_{j^*m^*}$ is the one defined on the optimal input variable $j^* = \arg\max_j J_j$ that maximizes the original criterion $m^* = \arg\max_m J(b_{j^*m})$.

In this context (14.7) it can be useful to define an additional ("central") subbox associated with real valued inputs x_j

$$b_{jc} = \{\mathbf{x} \,|\, x_{j(\alpha)} \leq x_j \leq x_{j(1-\alpha)}\}. \tag{14.8}$$

Including b_{jc} along with $\{b_{j-}, b_{j+}\}$ (7.5) in (14.7) makes the procedure sensitive to input variables x_j upon which the target function (within the current box) has a strong symmetric convex dependence, for example $f(\mathbf{x}) = x_j^2$, $-1 \leq x_j \leq 1$. In this case deleting either extreme subbox $b = b_{j\pm}$ (7.5)

decreases the mean in the remaining box $\bar{y}_{B-b} < \bar{y}_B$. For this reason, not including b_{jc} (14.8) in (14.7) will likely cause this x_j to be ignored for peeling, perhaps in favor of an irrelevant input that has no effect on the output y ($\bar{y}_{B-b} \simeq \bar{y}_B$). Note that b_{jc} is only used in (14.7) to help evaluate the importance of each real valued input variable; it is never eligible for actual deletion.

15. Marketing data

In this section the use of PRIM is illustrated on marketing data. It consists of $N = 9409$ questionnaires (502 questions) filled out by shopping mall customers in the San Francisco Bay area [Impact Resources, Inc. Columbus, OH (1987)]. The first 14 questions, relating to demographics, are listed in Table 15.1. These are seen to be a mixture of real and categorically valued variables. There are many missing values. The other 488 questions are concerned with various consumer behaviors. These can be used as output variables y to identify the demographics (input variables $\mathbf{x}$) of those respondents who engage in particular behaviors. Here we choose for illustration three characteristic behaviors of one of the authors (JHF).

Table 15.1

Input variables for the marketing data.

Var.	Demographic	No. of values	Cat.
1	sex	2	*
2	marital status	5	*
3	age	6	
4	education	6	
5	occupation	9	*
6	income	9	
7	years in Bay Area	5	
8	married - dual incomes	2	*
9	number in household	9	
10	number in household < 18	9	
11	householder status	3	*
12	type of home	5	*
13	ethnic classification	8	*
14	language in home	3	*

The first behavior examined is frequency of air travel as characterized by number of round trip flights per year. The global mean value over the entire data base is $\bar{y} = 1.7$. The first two boxes induced by PRIM are

y = number of flights/year; $\quad \bar{y} = 1.7$

$B_1 : \bar{y}_1 = 4.2, \quad \beta_1 = 0.08$

education $\geq$ 16 years
occupation $\in$ {professional / managerial, sales worker, homemaker}
income > \$50K, & $\neq$ missing
number of children (< 18) in home $\leq$ 1

$B_2 : \bar{y}_2 = 3.2, \quad \beta_2 = 0.07, \quad$ Dissimilarity: $D(B_1, B_2) = 0.14.$

education > 12 years, & $\neq$ missing
income > \$30K, & $\neq$ missing

$18 <$ age < 54
married / dual incomes $\in$ {single, married–one income}

The box "dissimilarity" diagnostic $D(B_1, B_2)$ (16.5) is described in Section 16.3. The first box identifies the demographics of a 8% market segment that averages 4.2 flights per year, and the second box another 7% segment, distinct from the first, with almost double the global average of 1.7. These boxes verify intuition; there are no real surprises revealed by these rules.

The next behavior examined is that of owning a pet. Fifty two percent of the respondents indicated that they had a pet; the odds of a randomly selected person in the sample owning a pet are roughly 1/1. The first two boxes induced by PRIM are

$y = 1$(have a pet). $\quad \bar{y} = 0.52.$

$B_1 : \bar{y}_1 = 0.80, \qquad \beta_1 = 0.17$

age ≤ 44
education ≤ 14 years
live in Bay Area ≥ 4 years
home $\in$ {house, mobile}
ethnic class $\in$ {Native American, East Indian,White, missing}

$B_2 : \bar{y}_2 = 0.76, \qquad \beta_2 = 0.08, \qquad$ Dissimilarity: $D(B_1, B_2) = 0.44.$

number of children (< 18) in home > 0
household status $\in$ {own, live with parents, missing}
ethnic class $\in$ {Native American, East Indian,White, missing}

PRIM identified two market segments collectively covering 25% of the entire sample for which the odds of owning a pet are roughly 5/1. Perhaps a surprise here is the importance of ethnic class in determining likelihood of pet ownership.

The third example relates to radio listening habits, specifically inclination to listen to KGO-AM

$$y = \text{ listen to KGO-AM } = \begin{cases} 0.0 & \text{never} \\ 0.25 & \text{occasionally} \\ 1.0 & \text{regularly.} \end{cases}$$

The global average is $\bar{y} = 0.10$. The first two rules induced by PRIM are

$B_1 : \bar{y}_1 = 0.23, \qquad \beta_1 = 0.14$

age ≥ 35
live in Bay Area ≥ 10 years, & $\neq$ missing
number of children in home ≤ 1
householder status $\neq$ rent
ethnic class $\notin$ {Black, Pacific Islander}
type of home $\in$ {house, condo}

$B_2 : \bar{y}_2 = 0.20, \qquad \beta_2 = 0.09, \qquad$ Dissimilarity: $D(B_1, B_2) = 0.77.$

sex = male
marital status $\in$ { married, divorced or separated, missing}
occupation $\in$ {professional / managerial, sales worker, retired}

These first two boxes cover a 23% market segment and identify demographics for which KGO-AM has over twice the listener share than in the entire sample.

16. Diagnostics

The basic output of the PRIM procedure is a set of rules (5.5) defining a series of boxes (5.1). They can be used to infer simultaneous values of particular input variables that relate to large values of the output variable. These rules also define respective subsets of the data, namely those observations that lie within each box B_k

$$\{y_i, \mathbf{x}_i \,|\, \mathbf{x}_i \in B_k\}. \tag{16.1}$$

Summary statistics based on these observations (16.1) can be used to provide further insight as to the nature of the induced region (5.1).

16.1. Sensitivity analysis

An important ingredient of any data analysis is determining the sensitivity of the objective to the induced parameter values. Here the objective is (high) box mean (7.1) and the parameters are the box boundaries (5.5). This sensitivity can be inferred from the rate of change of the target mean at the boundary values on each input variable defining the solution for each box. Let B be a solution box and B_{-j} the corresponding (larger) box with the jth input variable x_j removed from its definition(10.1). The function

$$\bar{f}_j(x_j) = E[y \,|\, x_j \;\&\; \mathbf{x} \in B_{-j}] \tag{16.2}$$

represents the target mean for each value of x_j, given the solution boundary values on the other variables $\{x_{j'}\}_{j' \neq j}$ defining the box (5.5). The slope of this function $d\bar{f}_j/dx_j$ at the x_j box boundaries provides a measure of the sensitivity of the box mean to those particular boundary values. For real–valued variables an estimate $\hat{f}_j(x_j)$ of (16.2) can be obtained by smoothing the output variable y on x_j for those observations contained in the enlarged box B_{-j}

$$\hat{f}_j(x_j) = \text{smooth}(y_{ij} \text{ vs. } x_{ij} \mid \mathbf{x}_i \in B_{-j}). \tag{16.3}$$

For categorical variables, (16.3) is simply the output mean for each of its values, $x_j \in S_j$, in B_{-j}.

Figure 16.1 shows plots of (16.3) for the four variables defining the first box B_1 of the frequent flyer example (y = number of flights per year) of Section 15. (The input variable values here are different than those indicated in Section 15 since each value in the data set represents a code for the corresponding actual value.) For the categorical variable (x_j = occupation - upper right frame) the values of (16.3) are represented by vertical bars. Associated with each bar are two quantities shown below it. The first is the value of x_j with "M" indicating missing value. The second is the fraction (nearest percent) of the observations in the corresponding box B_{-j} having that particular value of x_j. The horizontal line along the top of the plot is the output mean in the solution box B, provided for reference. The shading (or not) of each bar indicates whether (or not) its corresponding value is in the set s_{jk} (5.5) defining the box B. Here the values occup $\in \{1, 2, 5\}$ define the box on this variable. The sensitivity of the solution box B to each categorical value is reflected by the height of its bar relative to the horizontal line, and its relative support (percent). Here one sees that enlarging the box B by including additional values would reduce its mean. The box mean is least sensitive to inclusion of values $x_j \in \{6, 7, \text{M}\}$ which have values of (16.3) closest to the box mean. However, these values comprise only a small fraction of the data so box support is not significantly increased. Of those values defining the box, the exclusion of $x_j = 5$ would increase the box mean with a moderate reduction in its support. Excluding either of the other two values $x_j \in \{1, 2\}$ would substantially reduce the solution mean and support.

The other three frames of Fig.16.1 represent the orderable (real) valued variables defining the solution box. The line represents the smooth (16.3) for each, with the points representing the (here) discrete values of each corresponding variable. The box boundaries on each respective variable are

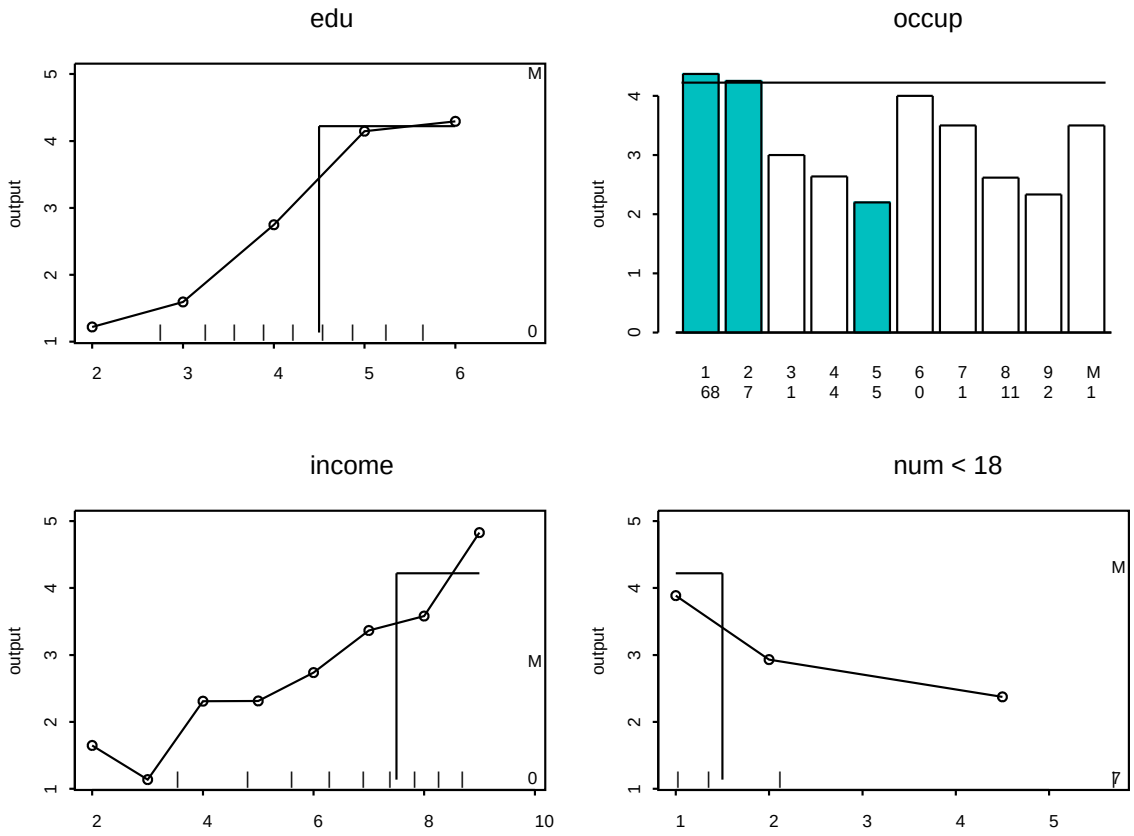

Figure 16.1: Sensitivity plots for the first box of the frequent flyer example. These can be used to judge the sensitivity of the box mean to the values of the induced boundaries, as well as suggest potential modifications to the boundaries.

indicated by the vertical lines. The horizontal line between the boundaries on each plot is the solution box B mean. Here each variable happens to define only one boundary with the other one being implicitly defined by the corresponding upper or lower limit of the plot. The "M" at the right of each plot represents (16.3) for missing values and the number below it shows the fraction of x_j = missing (to the nearest 10%) for observations in the enlarged box B_{-j}. The small vertical "hash" marks along the bottom of each plot delineate the deciles of the distribution of x_j for observations in B_{-j}. From Fig.16.1 one sees that the output mean in the solution box B is least sensitive to small changes in the income boundary and correspondingly much more sensitive with respect to the other two orderable variables (education and number of children in the household with age < 18). Note that for the latter variable, approximately 70% of its values in B_{-j} are missing ("7" in lower right corner). These mostly represent the value zero which was coded as missing for this variable in the data set.

As indicated by this example, sensitivity analysis like that provided in Fig.16.1 can aid the user in interpreting the solution produced by PRIM, and perhaps also in making modifications to it. With an interactive implementation the user could optionally modify the box boundaries based on information provided by the sensitivity plots. Corresponding sensitivity plots can then be produced for the modified box, perhaps suggesting further changes. This can be iterated until the user is satisfied with the box boundaries on all variables. Such an implementation essentially permits the user to manually execute the peeling/pasting strategy, starting with the PRIM solution. As with other aspects of the PRIM procedure, this permits the full incorporation of user judgement and domain knowledge.

16.2. Relative frequency plots

The rules (5.5) provide a definition of each box B_k. However, care is required in interpreting each of these definitions as a complete or unique description of B_k. Other descriptions, based on different input variables, may lead to very similar boxes in terms of the actual data points (16.1) that are covered. This is caused by collinearity among the input variables. As discussed in Section 10 high correlation between two (or more) input variables within a box can produce ambiguity in its definition; appropriately restricting the range of values of either variable (or both) gives rise to similar subsets of observations. One of the goals of the redundant variable elimination strategy is to select the best (most highly restrictive) variable within each such highly correlated group of inputs.

From the perspective of interpretation it is important to be aware of possible alternative definitions for each induced box B_k. The rule (5.5) defines the data (16.1) within the box. This data can be used to compare the relative frequency distribution of values of each input variable x_j within the box $p_j(x_j \,|\, \mathbf{x} \in B_k)$ to that over the entire data sample $p_j(x_j)$. Input variables (other than those used to define the box) for which the former distribution is more highly peaked than the latter, represent candidates for alternative definitions. Like the input variables that explicitly define B_k (5.5) the range of values of these other variables is also restricted within the box.

There are a variety of ways to compare two distributions. One possibility is through their ratio

$$r_{jk}(x_j) = p_j(x_j \,|\, \mathbf{x} \in B_k) \big/ p_j(x_j). \tag{16.4}$$

Note that the support of the denominator is always greater than that of the numerator. A uniform distribution for $r_{jk}(x_j)$ implies that x_j is totally irrelevant to the definition of B_k; the relative frequency of its values is the same inside or outside of the box. Departures from uniformity indicate association with the box definition. A highly peaked distribution for $r_{jk}(x_j)$ implies that the input x_j is highly relevant to the definition of B_k whether or not it is one of the defining variables (5.5).

Figure 16.2 shows relative frequency ratio plots $\{r_{j1}(x_j)\}_1^{14}$ (16.4) for the first box B_1 of the frequent flyer example (y = number of flights per year) of Section 15. The last bin of each plot represents the value x_j = missing. The other bins show the respective $r_{j1}(x_j)$ for the non-missing values. The first frame $r_{11}(x_1)$ shows that gender is not highly relevant to the definition of B_1; men ($x_1 = 1$) are slightly over–represented among these frequent fliers. The second plot shows that widowed people ($x_2 = 4$) are much less likely to be among these frequent flyers that in the total sample. The plot for x_3 (age) indicates that young [$x_3 \leq 2$ (24 years old)] and older [$x_3 \geq 7$ (55 years old)] people are highly under–represented in B_1. The distributions of x_4, x_5, and x_6 (education, occupation, and income) reflect

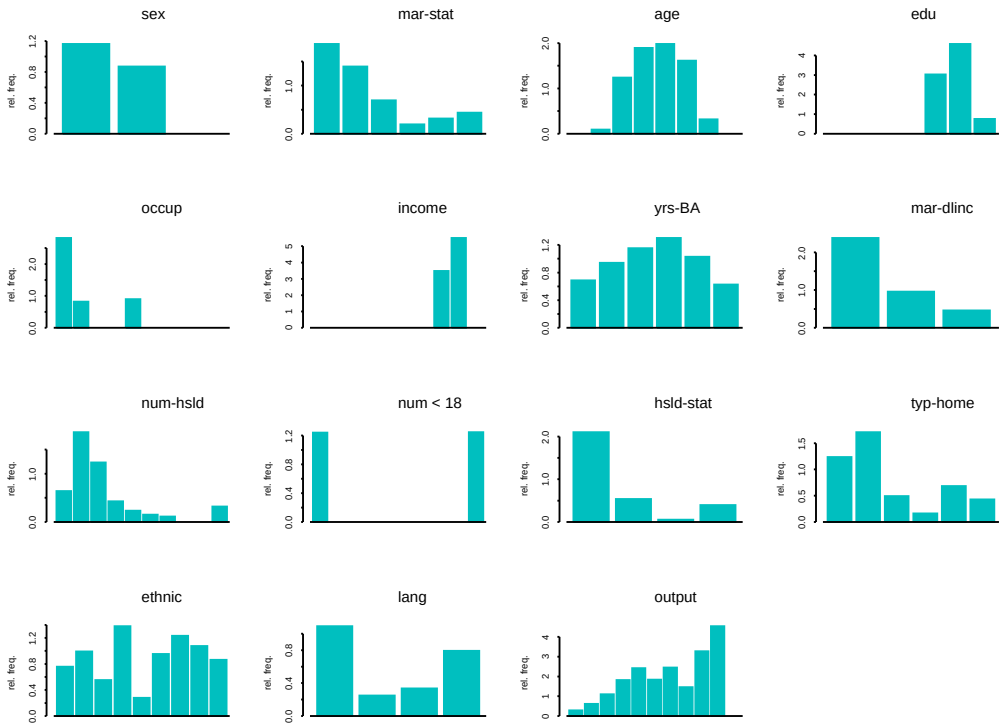

Figure 16.2: Relative frequency ratio distributions for the first box of the frequent flyer example. Shown are histograms of the frequency of values for each variable for observations in the box, relative to those in the entire data set. These can be used to assess the location of the box with respect to the values of all input variables, including those that do not appear explicitly in the box definition.

the box definition (Section 15). The plot of $r_{51}(x_5)$ shows that $x_5 = 1$ (occupation = professional / managerial) are almost three times more likely to be among the frequent flyers of B_1 than in the total sample. The distribution of $r_{91}(x_9)$ (number in household) reflects the box definition on number of children in the house less than 18 years old. The plot of $r_{11,1}(x_{11})$ (householder status) reflects that of $r_{31}(x_3)$ (age). Those people who live with parents or family ($x_{11} = 3$) tend to be younger ($x_3 \leq 24$ years old). The distribution of $r_{12,1}(x_{12})$ (type of home) indicates an over–representation of condominium owners ($x_{12} = 2$) and a dramatic under–representation of mobile home owners ($x_{12} = 4$) in B_1. The ethnic classification plot $r_{13,1}(x_{13})$ shows an over representation of East Indians ($x_{13} = 4$) and whites ($x_{13} = 7$), and an under–representation of African Americans ($x_{13} = 3$) and especially Hispanics ($x_{13} = 5$). This is also reflected in $r_{14,1}(x_{14})$; people who speak Spanish at home ($x_{14} = 2$) are less likely to be among the frequent flyers in B_1.

The last frame of Fig. 16.2 shows the distribution corresponding to (16.4) for the output variable, $r_{y1}(y)$. One sees that all values of y are represented in B_1. People who fly nine or more times per year ($y = 9$) are roughly four times more likely to be in B_1 than in the general population, whereas those odds are reversed for those who don't fly at all ($y = 0$).

The highly non-uniform distributions of $r_{j1}(x_j)$ (16.4) for many of the inputs not directly defining B_1 indicate the presence of strong associations among many of the inputs within the box. The choice here of x_4, x_5, x_6, and x_{10} to explicitly define B_1 is based on the data; they appear to be the most highly relevant as estimated by the redundant variable elimination strategy of Section 10. As with any estimation procedure, variance can cause errant choices among close alternatives. Also it is possible (if unlikely) that a highly relevant input variable can be completely masked by others highly correlated with it, and never be chosen for peeling. Although the patient strategy (Section 8) as well as multiple trajectories (Section 13) help mitigate this possibility, it cannot be completely eliminated. Examining the distributions of $\{r_{jk}(x_j)\}_1^n$ (16.4) can help diagnose such problems as well as giving an overall impression of the characteristics of the induced box. Of course, if two or more highly correlated variables appear from the data to be nearly equally relevant, only user judgement can ascertain which one (or both) actually are related to the output y.

For the marketing data (Table 15.1) there are at most nine different non–missing values for any input variable. With a sample size of $N = 9409$ the value of $r_{jk}(x_j)$ (16.4) can be reliably estimated for each individual value of x_j. In many data sets real–valued variables can assume a large number of distinct values. The geology data discussed in Section 11 is an example. For such real–valued inputs x_j, plots of (16.4) analogous to Fig. 16.2 are made by histogramming the x_j-values using as bin boundaries the γ-quantiles of the distribution of x_j over the entire data set $p_j(x_j)$. The number of bins ($1/\gamma$) is chosen by the user based on sample size N and the degree of detail required.

Finally, it is important to remember that the plots in Fig. 16.2 do *not* represent the frequency distribution of the x_j-values $p_j(x_j \,|\, \mathbf{x} \in B_k)$ within the box B_k. Rather, they represent those frequencies *relative* to the corresponding frequencies $p_j(x_j)$ over the entire data set (16.4). For example, the value $y = 9$ is *not* the most frequent output value in B_1 (last frame). This value is very infrequent in the entire data set. The value $p_{y1}(9) = 4.2$ indicates that the output value $y = 9$ is 4.2 times more likely to appear in B_1 than in the data set as a whole. Plots of $p_j(x_j \,|\, \mathbf{x} \in B_k)$ can also serve as useful additional diagnostics as can statistics other than (16.4) for comparing it to $p_j(x_j)$.

16.3. Interbox dissimilarity

The covering procedure (Section 6) produces a sequence of boxes $\{B_k\}_1^K$ that cover a subregion R (5.1) of the input variable space. No constraints are placed on the relationships among the boxes. They can overlap or be disjoint, be close or far apart, as dictated by the nature of the target function. For example, if the target function is characterized by a single prominent mode, the covering procedure might produce a series of nested boxes, each successively induced box completely covering all of those induced before it. Alternatively, successive boxes might cover different "shoulders" of that mode producing a "cluster" of closely related boxes. If there were several different prominent modes, the boxes might divide into corresponding groups of nested/clustered sequences. Knowledge concerning relationships among the boxes can provide information about the modal structure of the

target function.

Relationships among the boxes can be deduced from their pairwise dissimilarities. The "dissimilarity" $D(B_k, B_l)$ between boxes B_k and B_l is defined as the difference between the support of the smallest box B_{kl} that covers both of them, and the support of their union

$$D(B_k, B_l) = \beta(B_{kl}) - \beta(B_k \bigcup B_l). \tag{16.5}$$

Here the support $\beta(B)$, of a box B, is defined as the fraction of observations in the entire data set (1.1) that it covers. The minimal covering box is defined as

$$\mathbf{x} \in B_{kl} = \bigcap_{j=1}^{n} [x_j \in s_j(k,l)] \tag{16.6}$$

where the value subsets $s_j(k,l) \in S_j$ are derived from those of the respective two boxes $\{s_{jk}\}_{j=1}^n$ and $\{s_{jl}\}_{j=1}^n$ (5.3). For real valued variables x_j, $s_j(k,l)$ is the smallest interval $[t_j^-(k,l),\, t_j^+(k,l)]$ that covers the respective two intervals for B_k and B_l on x_j (5.4)

$$t_j^-(k,l) = \min(t_{jk}^-, t_{jl}^-), \quad t_j^+(k,l) = \max(t_{jk}^+, t_{jl}^+). \tag{16.7}$$

For categorical variables x_j, the subset $s_j(k,l)$ is the union of the respective B_k and B_l subsets

$$s_j(k,l) = s_{jk} \bigcup s_{jl}. \tag{16.8}$$

The dissimilarity measure (16.5) assumes values in the semi–open interval $0 \leq D(B_k, B_l) < 1$. Although $D(B_k, B_l)$ is not strictly a distance, it retains the essential property that its value increases as the value subsets $\{s_{jk}\}_{j=1}^n$, $\{s_{jl}\}_{j=1}^n$ (real or categorical), defining the two boxes B_k, B_l, increasingly differ on each of the input variables x_j. Nested boxes will have zero dissimilarity, as will "adjacent" boxes that have contiguous intervals on one real variable x_j, and identical subsets on all other variables $\{s_{j'k} = s_{j'l}\}_{j' \neq j}$. Other configurations will produce larger dissimilarity with the highest values occurring when the two boxes are defined by highly disparate value subsets (5.5) on one, or especially several, input variables. Note that two boxes can have non–zero intersection $[\beta(B_k \bigcap B_l) \neq 0]$ and still be highly dissimilar, so long as they are defined by very different subsets on the non–intersecting variables. Also note that box dissimilarity is defined in terms of the data distribution. Two boxes can be defined in terms of very different sets of input variables and still be quite similar, if the two sets are highly correlated.

Dissimilarities (16.5) are shown between the two boxes induced for each marketing data example in Section 15. For the first (frequent flyer) the two boxes are fairly similar $[D(B_1, B_2) = 0.14]$ indicating that both boxes represent similar market segments. For the second example (have a pet) the two induced boxes are moderately dissimilar $[D(B_1, B_2) = 0.44]$, representing somewhat different market segments. The two boxes in the third example (KGO-AM) are very different $[D(B_1, B_2) = 0.77]$. They represent very different demographic groups who are inclined to listen to KGO-AM radio.

17. Scaling to very large databases

The PRIM procedure is memory based. All of the data (1.1) is presumed to be stored in random access memory (RAM). While the RAM capacity of present day computers is sufficient for many applications, and is steadily growing, there are some data mining applications for which all of the relevant data can reside only on secondary (disk) storage. Although one could envision a disk–based implementation of PRIM, there are simple strategies to enable it to profitably use all of this data in a RAM based context.

The strategy employed depends on the (approximate) support β_0 (7.4) required for the induced boxes. If this is not very small ($\beta_0 \gtrsim 0.01$) then a modification of the bumping strategy (Section 13) can be employed. Successive trajectories (9.1) are induced on independent randomly selected subsets of the entire data base. The size of each subset is determined by available RAM. The multiple trajectory

strategy discussed in Section 13 is then employed to select the best box among all of those produced by the independently induced trajectories. This approach has a natural parallel implementation.

If very small supports ($\beta_0 \lesssim 0.01$) are desired, then a sequential sampling strategy can be used. As above, random subsamples are drawn from the data base. However, each successive subsample is required to be inside the box induced from the previous subsample. PRIM is applied to each successive sample with β_0 chosen to be relatively large ($\beta_0 \gtrsim 0.01$). Applying this procedure m times produces boxes with support β_0^m. For large enough data bases this sequential strategy can be coupled with the parallel one discussed above. These RAM based approaches presume a data base implementation that supports rapid selection of random subsets.

18. Competitors

Rule induction has a long history in the machine learning and statistics literatures. Although not specifically aimed at function optimization, many existing procedures can be applied in that context, at least in special cases. In this section we qualitatively compare PRIM to some of the more popular rule induction methods. A quantitative comparison with one (CART) is provided in Section 19.

18.1. Covering algorithms

When the output variable y assumes only two values (e.g. $y \in \{0, 1\}$), PRIM can be viewed as an inductive learning method where the two values respectively represent negative and positive instances of a target concept to be learned. In this context PRIM shares many of the characteristics of other machine learning algorithms that learn disjunctive sets of propositional rules through sequential covering. These include CN2 [Clark and Niblett (1989)] and propositional versions of FOIL [Quinlan (1990)] such as RIPPER [Cohen (1995)]. The principal difference is the search strategy used for examining the space of possible preconditions to construct each rule (box). These other procedures all use very greedy strategies, especially with categorical variables. As discussed in Section 8.1 this can limit performance in many situations and is the primary motivation for the development of the patient strategy employed by PRIM. A less important difference is in the representation of the induced rules. Each box induced by PRIM (5.5) can involve implicit disjunctions

$$x_j \in s_{jk} = \bigcup_{z_l \in s_{jk}} (x_j = z_l).$$

Most other methods produce purely conjunctive rules involving only equality constraints on the values of each categorical variable. Other differences include the strategy for missing values (Section 12) and the use of multiple trajectories (Section 13) instead of explicit backtracking.

18.2. Decision tree induction

In terms of actual application, procedures most competitive with PRIM are likely to be decision tree induction techniques such as CART [Breiman et al. (1984)] and C4.5 [Quinlan (1994)]. These methods produce a set of disjoint rules that collectively cover the entire input space through recursive partitioning. Each successive partition ("split") is induced by a condition $x_j \in s_{jk}$ so that each rule in the final covering set (terminal nodes) takes the conjunctive form given in (5.5). Although the goal of these procedures is accuracy of approximation everywhere in the input space, rather than explicit optimization, one can examine the rules associated with the highest predicted output values. These can be interpreted in an analogous manner to those induced by PRIM.

The major differences between PRIM and decision tree induction methods are the use of covering rather than partitioning to produce rule sets, and patient rather than greedy strategies to induce the individual rules. The latter issue is discussed in Section 8. The relative merits of covering versus partitioning are issues of debate in the machine learning literature [see Mitchell (1997)]. Rules produced by covering are more "expressive" than those produced by partitioning since they are induced independently of each other. Partitioning rules are forced to share many common conjunctive statements

$x_j \in s_{jk}$. Thus, covering tends to produce fewer rules, each of a simpler nature (fewer conjunctions) and are thereby more interpretable. This has motivated post-processing strategies such as C4.5rules [Quinlan (1994), (1995)] that attempt to simplify decision tree rule sets. For greedy strategies it is not clear whether the increased expressiveness of covering rules translates to higher accuracy; this is likely to be problem dependent. In the case of patient rule induction, covering seems to be the more natural alternative; a straightforward patient implementation based on partitioning does not appear to be obvious.

18.3. User involvement

A major difference between PRIM and its predecessors is the involvement of user judgement as an integral part of the rule induction process (Sections 9, 10, 13 and 16.1). The purpose of the exercise is presumed to be descriptive data analysis where specific goals are highly problem dependent. The PRIM procedure provides natural interfaces for user guidance towards individual goals, as well as for incorporating domain knowledge to improve accuracy. The user is given the opportunity to customize individual rules to produce the smallest simplest set consistent with programmatic needs. This is in contrast with other rule induction methods where all such trade–offs are automatically made by thresholding mathematical criteria. In these (automatic) settings the user assumes a passive role of simply inspecting the produced output.

19. PRIM versus CART

Among the competitors to PRIM the one using the least greedy strategy is CART [Breiman et al. (1984)]. On average, each conjunctive constraint $x_j \in s_{jk}$ defining the final rule set (5.5) (terminal nodes) selects one half of the data (binary split) at each iterative step, for both real and categorical variables. This can be viewed as a "semi"-greedy strategy when compared to most other methods that fragment the data much more severely. Also, CART produces rules that are identical in form (5.5) to those of PRIM, and it is directly equipped to handle real valued output variables (regression trees). In this section we compare the performance of CART and PRIM on several data sets from the perspective of function maximization. The first is synthetic data deliberately designed to illustrate the advantage of the patient strategy employed by PRIM. The others are the two geology and three marketing examples discussed respectively in Sections 11 and 15.

As with PRIM's other competitors, CART offers no user control over the mean–support trade-off of the rules it produces. Therefore, to make the results of the two procedures comparable the following procedure was employed. CART was first applied to the data set. Its best J rules (terminal nodes) in terms of highest predicted output value were identified. PRIM was then applied to the same data set to sequentially induce J boxes through the covering technique (Section 6). The mean–support trade–off of each box was chosen to match either the mean or support of the corresponding CART rule, depending on which one could be matched most closely. The relative power of the respective rules can then be compared in terms of the corresponding unmatched quantity. If the supports are similar then the one with the higher mean is better; if their means are similar the one with the larger support is better. In all cases the data was divided into the same learning and test sets (Section 9) for both procedures. The peeling fraction α (7.5) was taken to be 10% ($\alpha = 0.1$) for PRIM.

19.1. Synthetic example

The synthetic data consisted of $N = 10000$ observations with $n = 10$ input variables randomly drawn from a uniform distribution $\{x_j \sim U[-1, 1]\}_1^{10}$. The target function was taken to be

$$y = f(\mathbf{x}) = \prod_{j=1}^{J} x_j. \tag{19.1}$$

This target has 2^{J-1} maxima of equal value. Here we illustrate with $J = 3$ so that there are 4 maxima at alternating corners of the cube defined by the first three input variables. Table 19.1 compares the performance of CART and PRIM on this problem.

Table 19.1
Performance on the synthetic data example.

CART			PRIM		
k	$\bar{y}_k$	$\beta_k(\%)$	k	$\bar{y}_k$	$\beta_k(\%)$
1	0.29	1.4	1	0.28	4.2
2	0.24	1.3	2	0.31	3.3
3	0.19	1.4	3	0.32	3.5
4	0.14	1.9	4	0.34	3.1

Coverage ratio = 3.9

The left subtable shows the mean $\bar{y}_k$ and support β_k ($\times 100$) for the best four CART rules as averaged over ten (randomly) replicated data sets. The right subtable shows these quantities for the corresponding PRIM rules averaged over the same set of replications. The quantity "coverage ratio" is intended as a single summary measure of relative performance. It is the ratio of the "coverage"

$$C = \sum_{k=1}^{4} (\bar{y}_k - \bar{y}) \cdot \beta_k \tag{19.2}$$

of PRIM to that of CART. Here $\bar{y}$ is the global mean (1.8). In this sense (19.2) PRIM's coverage is (on average) almost four times that of CART for this problem.

This example (19.1) is intended to emulate situations in which the initial set of splits (CART) or peels (PRIM) are not well defined. In this case the target function provides no information on how to choose (at minimum) the first two. Splits/peels are selected randomly until two of the first three inputs have been selected. From that point on there is information to direct further splitting/peeling. Here PRIM dramatically out–performs CART owing to its patient strategy. Because each peel removes only 10% of the data, there tends to be much more data remaining when the target function structure is "revealed", allowing subsequent peels to increase target mean. With CART there is typically much less data left at the corresponding point along each branch of the decision tree.

PRIM found each of the target's four maxima with its first four rules on all ten trials. The reason why later rules do better than earlier ones here is a consequence of the covering strategy. Removing each successive maximum reduces the target symmetry over the remaining data. This allows its structure to be uncovered earlier in the peeling sequence.

The difficulty of this problem represents a limit for CART. Making the problem more difficult by increasing the number of inputs to $n = 20$, or reducing the sample size to $N = 1000$, or setting $J = 4$ in (19.1) to produce eight maxima, causes CART not to produce a tree. It fails to detect sufficient structure in the target function and its pruning algorithm subsequently removes all splits. In all of these more difficult cases PRIM continued to find all the target maxima with its first 2^{J-1} boxes.

19.2. Data examples

As noted, the synthetic example (19.1) was specifically contrived to exploit PRIM's advantages. One would not expect such large improvements over CART in most situations. Tables 19.2 - 19.6 show corresponding results to those of Table 19.1 (three rules only) for the two geology examples in Section 11 and the three marketing examples in Section 15 respectively.

Table 19.2
Performance on the geology example, ancient stable shields.

	CART			PRIM	
k	$\bar{y}_k$	$\beta_k(\%)$	k	$\bar{y}_k$	$\beta_k(\%)$
1	0.91	11.0	1	0.94	11.0
2	0.87	5.3	2	0.87	6.0
3	0.79	4.3	3	0.80	6.6

Coverage ratio = 1.20

Table 19.3
Performance on the geology example, young orogenic belts.

	CART			PRIM	
k	$\bar{y}_k$	$\beta_k\,(\%)$	k	$\bar{y}_k$	$\beta_k(\%)$
1	0.64	2.1	1	0.67	2.2
2	0.12	3.6	2	0.25	3.5
3	0.10	2.9	3	0.09	3.4

Coverage ratio = 1.34

Table 19.4
Performance on the marketing example, number of flights per year.

	CART			PRIM	
k	$\bar{y}_k$	$\beta_k\,(\%)$	k	$\bar{y}_k$	$\beta_k(\%)$
1	5.0	1.9	1	5.1	2.3
2	3.6	2.5	2	3.6	5.1
3	2.5	2.7	3	3.0	5.6

Coverage ratio = 1.87

Table 19.5
Performance on the marketing example, have a pet.

	CART			PRIM	
k	$\bar{y}_k$	$\beta_k\,(\%)$	k	$\bar{y}_k$	$\beta_k(\%)$
1	0.81	4.5	1	0.84	7.8
2	0.75	16.0	2	0.72	22.7
3	0.71	2.4	3	0.69	2.3

Coverage ratio = 1.38

Table 19.6
Performance on the marketing example, listen to KGO-AM.

	CART			PRIM	
B	$\bar{y}_B$	$\beta_B\,\%$	B	$\bar{y}_B$	$\beta_B\,\%$
1	0.25	6.1	1	0.23	14.0
2	0.17	7.9	2	0.17	11.0
3	0.16	5.8	3	0.13	10.0

Coverage ratio = 1.44

The rules induced by PRIM are seen to be either comparable to or better than those of CART, sometimes substantially so. This is especially the case for the marketing data. The coverage ratio

averaged over these five examples is 1.45. PRIM's coverage is on average 45% higher than that of CART.

In all but one example (Table 19.3) CART produced very large trees (50 - 200 terminal nodes). Most of its rules were far more complex than those produced by PRIM. Techniques analogous to C4.5rules [Quinlan (1994), (1995)] might help simplify the CART rules in this context. CART was used with all parameter settings at their default values. These are presumably tuned for regression analysis where the goal is target function approximation over the entire input space. It is conceivable that there are other settings that are more appropriate for optimization. Finally, it should be noted that the rules summarized in Tables 19.2 - 19.6 are generally not the same — nor as good — as the corresponding ones presented in Sections 11 and 15. The mean-support trade-offs were selected here to align with those produced by CART, rather than to produce the most favorable joint (mean–support) values.

20. Summary

PRIM is intended as an addition to the data analyst's tool kit, to be used when the goal is either explicit (Section 3) or implicit (Section 4) optimization. Its distinguishing characteristics include patient peeling/pasting (Sections 7 and 8) coupled with multiple trajectories (Sections 13 and 14) to enhance power and stability, and intimate user involvement in the model selection process (Sections 9, 10, 13 and 14.1). It tends to produce parsimonious interpretable descriptions of the structure it uncovers (Sections 11, 15, and 16), and on the problems considered here (Section 19) exhibits performance superior to comparable procedures such as CART that are intended for function approximation. The extent to which this performance gain generalizes to other situations remains to be established. As with all learning procedures, relative performance will likely be problem dependent.

21. Acknowledgments

We thank Bill Griffin and Impact Resources, Inc. for providing respectively the geology and marketing data bases used for illustration. We especially acknowledge the contribution of Art Owen who suggested the specific multiple trajectory ("thinning") strategy described in Section 13.

The work of Jerome H. Friedman was partially supported by the Department of Energy under contract DE–AC03–76SF00515, and by the National Science Foundation under grants DMS–9403804 and DMS–9704431. This research was also supported in part by the Australian Government International Science & Technology Grant for research collaboration: *Statistical Modelling and Prediction of the Structure of the Earth's Upper Mantle: a Framework for Mineral Exploration.*

References

[1] Barnett, V. (1976). The ordering of multivariate data (with discussion). *J. Roy. Statist. Soc. A* **139**, 318-354.

[2] Bishop, C. M. (1995). *Neural Networks for Pattern Recognition.* Oxford University Press.

[3] Breiman, L. (1996). Bagging predictors. *Machine Learning* **24**, 123-140.

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

[5] Clark, P. & Niblett, R. (1989). The CN2 induction algorithm. *Machine Learning* **3**, 261-284.

[6] Cohen W. W. (1995). Fast efficient rule induction. In *Machine Learning: Proceedings of the Twelfth International Conference*, lake Tahoe, CA (115-123). Morgan Kaufmann.

[7] Donoho, D. and Gasko, M. (1992). Breakdown properties of location estimates based on halfspace depth and projected outlyingness. *Annals of Statistics* **20**, 1803-1827.

[8] Efron, B. & Tibshirani, R. J. (1993). *An Introduction to the Bootstrap*. Chapman & Hall.

[9] Friedman, J. H. (1997). On bias, variance, 0/1-loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery* **1**, 55-77.

[10] Green, P. J. (1981). Peeling bivariate data. In *Interpreting Multivariate Data* (V. Barnett, ed.) Wiley.

[11] Griffin, W. L., Fisher, N. I., Friedman, J. H., Ryan, C. G., and O'Reilly, S. (1997). Cr-Pyrope garnets in lithospheric mantle. *J. Petrology* (submitted).

[12] Hall, P. (1989). On projection pursuit regression. *Annals of Statistics* **17**, 573-588.

[13] Lorentz, G. G. (1986). *Approximation of Functions*. Chelsea.

[14] Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill.

[15] Quinlan, J. R. (1990). Learning logical definitions from relations. *Machine Learning* **5**, 239-266.

[16] Quinlan, J. R. (1994). *C4.5: Programs for Machine Learning*. Morgan-Kaufmann.

[17] Quinlan, J. R. (1995). MDL and categorical theories (continued). In *Machine Learning: Proceedings of the Twelfth International Conference*, lake Tahoe, CA (464-470). Morgan Kaufmann.

[18] Ripley, B. D. (1996) *Pattern Recognition and Neural Networks*. Cambridge University Press.

[19] Rivest, R. L. (1987). Learning decision lists. *Machine Learning* **2**, 229-246.

[20] Tibshirani, R. J. and Knight, K. (1995). Model search and inference by bootstrap "bumping". Technical Report, University of Toronto.

[21] Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer.

[22] Wahba, G. (1990). *Spline Models for Observational Data*. SIAM.