PRIM (Patient Rule Induction Method)

Stephen Vardeman
Analytics Iowa LLC
ISU Statistics and IMSE

"PRIM"

"PRIM" is another rectangle-based method of making a predictor on $\Re^p$. The language seems to be "patient" as opposed to "rash." "Rule induction" is perhaps "predictor development" or more likely "conjunctive rule/rectangle development." In spirit it is a type of "bump-hunting."

A series of rectangles and predictor

For a series of rectangles (or boxes) in p-space

$$R_1, R_2, \ldots, R_l$$

one defines a predictor

$$\hat{f}_l(\mathbf{x}) = \begin{cases} \overline{y}_{R_1} & \text{if } \mathbf{x} \in R_1 \\ \overline{y}_{R_2 - R_1} & \text{if } \mathbf{x} \in R_2 - R_1 \\ \vdots & \vdots \\ \overline{y}_{R_m - \cup_{k=1}^{m-1} R_k} & \text{if } \mathbf{x} \in R_m - \cup_{k=1}^{m-1} R_k \\ \vdots & \vdots \\ \overline{y}_{\left(\cup_{k=1}^{l} R_k\right)^c} & \text{if } \mathbf{x} \notin \cup_{k=1}^{l} R_k \end{cases}$$

The boxes or rectangles are defined recursively in a way intended to catch "the remaining part of the input space with the largest output values."

A first rectangle

That is, to find R_1

1. identify a rectangle

$$l_1 \le x_1 \le u_1$$
$$\vdots$$
$$l_p \le x_p \le u_p$$

   that includes all input vectors in the training set,
2. identify a dimension, j, and either l_j or u_j so that by reducing u_j or increasing l_j just enough to remove a fraction α (say $\alpha = .1$) of the training vectors currently in the rectangle, the largest value of

$$\overline{y}_{\text{rectangle}}$$

   possible is produced, and update that boundary of the rectangle,

A first rectangle cont.

3. repeat 2. until some minimum number of training inputs $\mathbf{x}_i$ remain in the rectangle (say, at least 10),
4. expand the rectangle in any direction (increase a u or decrease an l) adding a training input vector that provides a maximal increase in $\overline{y}_{\text{rectangle}}$, and
5. repeat 4. until no increase is possible by adding a single training input vector.

This produces R_1. For what it is worth, step 2. is called "peeling" and step 4. is called "pasting."

Subsequent rectangles

Upon producing R_1, one removes from consideration all training vectors with $\mathbf{x}_i \in R_1$ and repeats 1. through 5. to produce R_2. This continues until a desired number of rectangles has been created. One may pick an appropriate number of rectangles (l is a complexity parameter) by cross-validation and then apply the procedure to the whole training set to produce a set of rectangles and predictor on p-space that is piece-wise constant on regions built from boolean operations on rectangles.