Generalized Additive Models, Recursive Partitioning Trees & PRIM

Multivariate Adaptive Regression Splines and HME
Linear Models

- Advantages
  - Global
  - Efficient
  - Interpretable

- Disadvantages
  - Non-linearity
    - GAMs address this
  - Interactions
    - RP Trees/PRIM/HME address these
    - MARS handles non-linearity and 2-way interactions
Generalized Additive Model

- Mean Function \( \mu(X) = E[Y \mid X] \)
- Link Function
  \[
g(\mu[X]) = \alpha + f_1(X_1) + \ldots + f_p(X_p)
  \]
  - \( f_j \)'s: unspecified smooth functions (non-parametric)

- Generalized Linear Model - Special case
  \[
f_j(X_j) = \beta_j X_j
  \]
Standard Link Functions

- **Identity:** $g(\mu) = \mu$
  - Used with normal error (linear) models

- **Logit:** $g(\mu) = \log[\mu/(1-\mu)]$
  - Used with 0-1 outcome data (Binomial family)

- **Probit:** $g(\mu) = \Phi^{-1}(\mu)$
  - Similar to Logit, but with shorter tails

- **Log:** $g(\mu) = \log(\mu)$
  - Used when outcomes are Poisson counts, or for log-linear models
Advanced Link Functions

- \( V \in \{1, \ldots, k, \ldots, K\} \) a categorical input
- \( W, Z \) additional continuous inputs
  - Semi-parametric Model: \( g(\mu) = X^T \beta + \alpha_k + f(Z) \)
    - Cox Model: \( g = \log \)
  - Simple Interaction: \( g(\mu) = f(X) + g_k(Z) \)
  - Full 2-way interaction: \( g(\mu) = f(X) + g(W,Z) \)
- Here \( f, g, g_k \) are estimated in a flexible manner based on penalized Residual SS
  - use some smoothing algorithm
    - E.g., Smoothing Splines
Fitting Additive Models

\[ Y = \alpha + \sum f_j(X_j) + \varepsilon \]

- Initialize:
  - Constant \( \alpha = \) average response value
  - Component functions \( f_j = 0 \)

- Cycle
  - Fit one component at a time to the residuals from the other components, using a smoother;
  - Normalize most recent component to average to 0;
  - Stop when all components converge within desired accuracy.
Backfitting Algorithm

- Initialize \[ \alpha = \frac{1}{N} \sum_{i=1}^{N} y_i; \hat{f}_j = 0, j = 1, \ldots, p \]
- Cycle: \( j = 1, 2, \ldots, p, \ldots, 1, 2, \ldots, p, \ldots \),

\[
\hat{f}_j \leftarrow S_j \left[ \left\{ y_i - \hat{\alpha} - \sum_{k \neq j} \hat{f}_k(X_{ik}) \right\}_{i=1}^{N} \right] \\
\hat{f}_j \leftarrow \hat{f}_j - \frac{1}{N} \sum_{i=1}^{N} \hat{f}_j(x_{ij})
\]

- Until the functions \( \hat{f}_j \) stabilize.
Fitting GAMs

- Instead of minimizing Residuals SS, subject to smoothing conditions, Maximize likelihood subject to constraints (penalized likelihood)
  - Newton-Raphson Method for generalized linear models becomes
    - Iteratively Re-weighted Least Squares
      - Fit a weighted linear regression to a working response variable on the covariates
      - Fitted regression yields a new value of the parameter estimates, in turn suggesting new working responses and weights
    - Iterate until convergence
Local Scoring Algorithm for Additive Logistic Regression

1. Initialize: \( \hat{\alpha} = \log \left[ \bar{y} / (1 + \bar{y}) \right] \); \( f_j = 0, j = 1, \ldots, p \)

2. Define: \( \hat{\eta}_i = \hat{\alpha} + \sum_j \hat{f}_j(x_{ij}); \hat{p}_i = 1/[1 + \exp(-\hat{\eta}_i)] \)

Iterate:

(a) Construct working target: \( z_i = \eta_i + \frac{(y_i - p_i)}{p_i(1 - p_i)} \)

(b) Construct weights: \( w_i = \hat{p}_i(1 - \hat{p}_i) \)

(c) Fit additive model to targets \( z_i \) with weight \( w_i \) using backfitting, giving new estimates \( \hat{\alpha}, f_j \), for all \( j \).

Stop when converged within specified accuracy.
E-mail/Spam Classification

Training Set
- 3065 e-mail messages to George Forman
- \( P = 57 \) quantitative predictors
  - Percent of words = each of 48 keywords
  - Percent of punctuation characters = ; ( [ ! $ #
  - Average length of sequence of capitals [CAPAVE]
  - Max length of sequence of capitals [CAPMAX]
  - Sum of entries used in CAPAVE [CAPTOT]

Test Set
- 1536 additional messages randomly chosen
E-mail/Spam Results

- **Smoothing**
  - Each predictor cubic spline smoothed to about 4 df
  - Test error
    - for smoothed predictors: 5.3%
    - For linear predictors: 7.6%

- **Info on 16 (out of 57) significant predictors**
  - Linear coefficient
  - Std error
  - Z-score
  - Non-linear p-value (test for non-linear effect)
    - All but 5 of the 16 have significant non-linear effects
    - Non-linear effect seemed to be due to discontinuity at 0
Figure 9.1: Spam analysis: estimated functions for significantly predictors. The rug plot along the bottom of each frame indicate the observed values of the corresponding predictor. For many of the predictors the nonlinearity picks up the discontinuity at zero.
Recursive Partitioning Trees

- For Regression
  - Approximates response by step function
    - Min Residual SS, the value of step function: Ave(y|x in subset m).
  - For each predictor $X_j$ find cut-point that minimizes the within group SS for the two resulting subsets
  - Choose best $X_j$ and make that cut
  - Repeat with each of the subset m
  - Continue until reach a minimum node size
  - Prune back the tree using
    - Cost complexity: For a subtree, Res SS + $\alpha |T|$
Figure 9.2: Partitions and CART. Top right panel shows a partition of a two-dimensional feature space by recursive binary splitting, as used in CART, applied to some fake data. Top left panel shows a general partition that cannot be obtained from recursive binary splitting. Bottom left panel shows the tree corresponding to the partition in the top right panel, and a perspective plot of the prediction surface appears in the bottom right panel.
RP Trees for Classification

- $\hat{P}_{mk}$: proportion of class $k$ on node $m$
- $k(m)$ be the majority class on node $m$

Instead of Within group SS, for each node, partition to minimize

- Misclassification Error: \( \frac{1}{N_m} \sum_{i \in R_m} I(y_i \neq k(m)) = 1 - \hat{P}_{mk}(m) \)
- Gini Index: \( \sum_{k \neq k'} \hat{P}_{mk} \hat{P}_{mk'} = \sum_{k=1}^{K} \hat{P}_{mk} (1 - \hat{P}_{mk}) \)
- Cross-entropy (deviance): \( \sum_{k=1}^{K} \hat{P}_{mk} \log(\hat{P}_{mk}) \)
Node Impurity Measures

Figure 9.3: Node impurity measures for two-class classification, as a function of the proportion $p$ in class 2. Cross-entropy has been scaled to pass through $(0.5, 0.5)$. 
Results for E-mail/Spam

Figure 9.4: Results for spam example. The green curve is the tenfold cross-validation estimate of misclassification rate as a function of tree size, with ± two standard error bars. The minimum occurs at a tree size with about 17 terminal nodes. The red curve is the test error, which tracks the CV error quite closely. The cross-validation was indexed by values of α, shown above. The tree sizes shown below refer to $|T_\alpha|$, the size of the original tree indexed by α.
Figure 9.5: The pruned tree for the spam example. The split variables are shown in blue on the branches, and the classification is shown in every node. The numbers under the terminal nodes indicate misclassification rates on the test data.
ROC Curves

- **Sensitivity vs Specificity**
  - Sensitivity: probability of predicting disease given true state is the disease
  - Specificity: probability of predicting non-disease given true state is non-disease

- Relative importance determines rule for classifying each terminal node.

- Can also use relative importance to weight observations when constructing the tree.
RoC Curves: E-Mail/spam

Figure 9.6: ROC curves for the classification rules fit to the spam data. Curves that are closer to the northeast corner represent better classifiers. In this case the GAM classifier dominates the trees. The weighted tree achieves better sensitivity for higher specificity than the unweighted tree. The numbers in the legend represent the area under the curve.
Classifying Biologically Active Groups of Chemicals

- Large Data Set: N~30,000; p~35,000
- Verducci, Fligner et al
- Recursive Partitioning (RP)
  - Split database sequentially according to the feature that maximizes difference in mean activity and/or proportion of actives
- RP + Simulated Annealing (RPSA)
  - Stochastic search for combinations of features that approximately optimize split
Recursive Partitioning
Applied to LNS-H23 activity in NCI database
RP parameters: max p-value = 0.01, min set size = 50
Recursive Partitioning

Advantages
- Useful for explaining complex, nonlinear response
- Handle very large descriptor sets with continuous, discrete, or categorical variables
- Handle very large data sets

Disadvantages
- Only optimizes one variable at a time
- Looks at few combinations of descriptors
- Most terminal nodes involve many negative descriptors
Stochastic Tree Search

- At each node, simulated annealing is used to find a combination of structural features

- Control parameters:
  - Number of features (descriptors)
  - Minimum node size
  - Maximum negative features
  - Number of tree levels

- Want to find local optima

- Modification -- drop certain features in the process
Stochastic Tree

<table>
<thead>
<tr>
<th>Node</th>
<th>Ave. pGI_{50}</th>
<th>Count</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7.35</td>
<td>51</td>
<td>oxetane, 3-oxy-; hdonor-path8-hdonor</td>
</tr>
<tr>
<td>2</td>
<td>7.49</td>
<td>54</td>
<td>benzene, 1-carbonyl, 4-(2-oxyethyl); hdonor-path8-pcharge</td>
</tr>
<tr>
<td>3</td>
<td>7.11</td>
<td>53</td>
<td>carbonyl, oxymethyl-; pyridine, 2-(alkenyl, cyc)-</td>
</tr>
<tr>
<td>4</td>
<td>6.66</td>
<td>52</td>
<td>oxepin, 3-oxymethyl-; alcohol, s-alkyl-</td>
</tr>
<tr>
<td>5</td>
<td>7.6</td>
<td>60</td>
<td>benzene, 1,3-dimethoxy-; cycloheptatriene, 1,3,5-</td>
</tr>
</tbody>
</table>

RP/SA parameters: min set size = 50, number of features in combination = 2.
**Patient Rule Induction Method**

Continuous Predictors

**Goal:** Isolate a single class with large proportion of observations of a given class

1. Start with entire training set (box)
2. Peal off a proportion $a$ from one face that most increases the proportion of desired class.
3. Repeat step 2 until some minimal number remain in the box.
4. Expand the box along any face that increases the proportion.
5. Use CV to trace back to best predictive box.
6. Remove the Box and repeat steps 2-5 until no more improvement in CV prediction.
Illustration of PRIM method

Figure 9.7: Illustration of PRIM algorithm. There are two classes, indicated by the blue (class 0) and red (class 1) points. The procedure starts with a rectangle (shown in orange) surrounding all of the data, and then peels away points along one edge by a prespecified amount in order to maximize the mean of the points remaining in the box. Starting at the top left panel, the sequence of peelings is shown, until a pure red region is isolated in the bottom right panel.
Box Mean Vs. Box Size

Figure 9.8: Box mean as a function of number of observations in the box.
End of Tuesday Lecture

- Next time: MARS and HME