Multivariate Adaptive Regression Splines

and

Hierarchical Mixture of Experts
Basis Elements of MARS

- Piecewise linear basis functions knotted at \( t \) and linked as a reflected pair
  - \((x-t)_+ = x-t\) if \( x > t \) (= 0 otherwise)
  - \((t-x)_+ = t-x\) if \( x < t \) (= 0 otherwise)
- Linear Splines

- For each input variable, form reflective pairs knotted at each of the observed values \( \{x_{ij} \mid i=1,\ldots,N\} \) of \( X_j, j=1,2,\ldots,p \)
Figure 9.9: The basis functions \((x - t)_+\) (solid red) and \((t - x)_+\) (broken green) used by MARS.
MARS Model Building

- Collection of $2Np$ basis functions

\[ \mathbb{C} = \{(X_j - t)_+, (t - X_j)_+ | t \in \{x_{1j}, \ldots, x_{Nj}\}; j = 1, \ldots, p\} \]

- Model

\[ f(X) = \beta_0 + \sum_{m=1}^{M} \beta_m h_m(X) \]

- Each of the $M$ basis functions $h_m$
  - Either a function in $\mathbb{C}$
  - Or a product of two or more such functions

- Given a choice for the $h_m$, the coefficients $\beta_m$ are estimated by minimizing the Residual SS
Figure 9.10: Schematic of the MARS forward model-building procedure. On the left are the basis functions currently in the model: initially, this is the constant function $h(X) = 1$. On the right are all candidate basis functions to be considered in building the model. These are pairs of piecewise linear basis functions as in Figure 9.9, with knots at all unique observed values $x_{ij}$ of each predictor $X_j$. At each stage we consider all products of a candidate pair with a basis function in the model. The product that decreases the residual error the most is added into the current model. Above we illustrate the first three steps of the procedure, with the selected functions shown in red.
Forward Selection

- Begin with constant function $h_0(x) = 1$.
  - Fit coefficient $b_0$ to minimize Residual SS
- At the first stage pick the best function $h_1(x)$ of the form
  $$ b_1(X_j - x_{ij})_+ + b_2(x_{ij} - Xj)_+ $$
  - Add this pair of basis functions to the current set $M$ of basis functions
- At the second stage select a new term of the form
  $$ \beta_3 h_\ell(X_j)(X_j - t)_+ + \beta_4 h_\ell(X)(t - Xj)_+ $$
  - for some $h_\ell(X)$ in $M$
- Continue until $M$ contains a pre-specified maximum number of basis terms.
  - We now have a large model that typically overfits the data
General Basis Selection Rule

- At each stage consider a new basis function pair
  - all products of a function $h_\ell$ in the model set $M$ with one of the reflected pairs in the basis set $C$.
- Added to the model is a term of the form
  - $\beta_{m+1} h_\ell(X) \cdot (X_j - t)_+ + \beta_{m+2} h_\ell(X) \cdot (t - X_j)_+$
  - with the $h_\ell$ in $M$, that produces the largest decrease in training error
- Note: the constant function $h_0$ is always in $M$
Product of two basis elements

Figure 9.11: The function $h(X_1, X_2) = (X_1 - x_{51})_+ \cdot (x_{72} - X_2)_+$, resulting from multiplication of two piece-wise linear MARS basis functions.
1. Remove the term whose removal causes the smallest increase in residual sum of squares.

2. Let $\lambda$ index the models as each term is successively deleted.

   Note: Typically not the same as the order in which the terms were added.

3. Use the Generalized Cross Validation criterion:

   $$GCV(\lambda) = \sum_{i=1}^{N} \frac{(y_i - f_{\lambda}(x_i))^2}{(1 - M(\lambda)/N)^2}$$

   where $M(\lambda)$ is the effective number of parameters.
Effective # of Parameters

- Includes:
  - Number \( r \) of terms (linearly independent basis functions) in the model, plus
  - Number of parameters used in selecting the optimal positions of the \( K \) knots
    - Guideline is a penalty \( c = 3 \) parameters per knot in piecewise linear basis function
    - Guideline is a penalty of \( c = 2 \) parameters per knot for additive models.

- Formula:
  \[
  M(\hat{\lambda}) = r + cK
  \]
Motivation

- Biggest advantage is local behavior
  - Since hinge basis functions are 0 over a part of their range, multiplication makes the product non-zero only over a small part of the feature space.
  - The result is to build up the regression surface parsimoniously.
- Computational advantage
  - Fits each potential basis function in $O(N)$ time:
    - For any input $X_j$, fit rightmost knot first [$O(N)$]
    - Move sequentially downward and update [$O(1)$]
  - Hierarchical interaction greatly limits the number of possible terms that need to be considered at any stage.
  - Do not allow same basis function more than once in a product
  - Order of interactions can be limited (e.g. only pairwise)
Spam Example

- GCV chooses a model size of 60 terms, which is roughly the smallest model giving optimal performance.
- Leading interaction terms involve ($, "remove"), ($, "free") and ("hp", CAPTOT)
- Overall test error rate is 5.5% (slightly worse than GAM, which gave 5.3%)
Spam Data: Test Error Rates for MARS

Figure 9.12: Spam data: test error misclassification rate for the MARS procedure, as a function of the rank (number of independent basis functions) in the model.
Simulation Study

- Scenario 1: Tensor product $p=2$
  \[ Y = (X_1-1)_+ + (X_1-1)_+ \cdot (X_2-.8)_+ + 0.12 \cdot \varepsilon \]

- Scenario 2: Tensor product $p=20$
  Scenario 1, but with 18 additional noise variables as inputs

- Scenario 3: Neural Network: Single hidden layer
  \[
  \ell_1 = X_1 + X_2 + X_3 + X_4 + X_5 \\
  \ell_2 = X_6 - X_7 + X_8 - X_9 + X_{10} \\
  \sigma(t) = 1/(1 + e^{-t}) \\
  Y = \sigma(\ell_1) + \sigma(\ell_2) + 0.12\varepsilon
  \]
Single Hidden Layer ANN

Figure 11.2: Schematic of a single hidden layer, feed-forward neural network.
Simulations Results for MARS

- Five simulation runs for each scenario
- Let \( \mathcal{S} \) denote a test set
- Let \( \mu(x) \) denote the true response at input \( x \)

\[
MSE_0 = \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} (\bar{y} - \mu(x))^2
\]

\[
MSE = \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} (\hat{f}(x) - \mu(x))^2
\]

\[
R^2 = \frac{MSE_0 - MSE}{MSE_0}
\]

Scenario 1: Tensor Product \( p=2 \)  
Mean \( R^2 = .97 \)  
S.E.=.01

Scenario 2: Tensor Product \( p=20 \)  
Mean \( R^2 = .96 \)  
S.E.=.01

Scenario 3: Neural Network  
Mean \( R^2 = .79 \)  
S.E.=.01
MARS for Classification

- For 2 classes
  - Can code Y as 0-1 and treat as continuous
  - Can replace Y by smoothed scores $S_\lambda Y$
- For $k > 2$ classes
  - “optimal scoring” of $Y^T S_\lambda Y$ where $Y$ is $N \times k$ matrix of class indicator variables
  - PolyMARS multiple logistic framework using a quadratic approximation to the multinomial log-likelihood
MARS vs CART

- Making the following changes converts the MARS procedure into CART (Recursive Partitioning Tree)
  - Replace the piecewise linear basis functions by step functions $I(x-t > 0)$ and $I(x-t < 0)$.
  - When a model term is involved in a multiplication by a candidate term, it gets replaced by the interaction (and hence is not available for further interactions)
- MARS Forward Selection is same as the tree-growing algorithm
Summary of MARS Features

- Separating relevant from irrelevant predictors
  - Large numbers of variables are examined using efficient algorithms, and all promising variables are identified.

- Transforming predictors exhibiting a nonlinear relationship with the target
  - Every variable selected for entry into the model is repeatedly checked for non-linear response. Highly non-linear functions can be traced with precision via essentially piecewise regression.

- Determining interactions between predictor variables
  - MARS repeatedly searches through the interactions allowed by the analyst.

- Unlike RP schemes, MARS models may be constrained to forbid interactions of certain types, thus allowing some variables to enter only as main effects, while allowing other variables to enter as interactions, but only with a specified subset of other variables.

- Handling missing values with new nested variable techniques
  - Certain variables are deemed to be meaningful (possibly non missing) in the model only if particular conditions are met (e.g., X has a meaningful non missing value only if categorical variable Y has a value in some range).

- Conducting extensive self tests to protect against overfitting
  - The user can choose to reserve a random subset of data for test, or use v-fold cross validation to tune the final model selection parameters.
Software

- Available in R from
  http://www.r-project.org/
- Free user-friendly trial copy available at
Hierarchical Mixture of Experts

- Tree with soft splits (assigning branching probabilities) at each non-terminal node (Gating network)
- Terminal nodes (Experts) contain models $P_\theta$ for $Y$, with different values of the parameter $\theta$
Hierarchical Mixture of Experts

Figure 9.13: A two-level hierarchical mixture of experts (HME) model.
Classification

- Top Gating Network “soft splits” into \( K \) possible groups
  - the probability of an input \( x \) falling into group \( j \) is

\[
g_j(x, \gamma_j) = \frac{e^{\gamma_j^T x}}{\sum_{k=1}^{K} e^{\gamma_k^T x}}, \quad j = 1, 2, \ldots, K
\]

- Note that for \( K=2 \) and \( \gamma_1 = +\infty \) for one of the \( x \) variables, this gives a hard split on \( x>0 \) vs \( x<0 \).
Gating Networks have a similar form:

\[ g_{\ell|j}(x, \gamma_j) = \frac{e^{\gamma_{j\ell}^T x}}{\sum_{k=1}^{K} e^{\gamma_{jk}^T x}}, \ell = 1, 2, ..., K \]
HME Models

- Terminal/Expert Node Contains a model for $Y$
  - Regression:
    - $Y = \beta_{jl} x + \varepsilon$
  - Classification
    - $\Pr(Y=1 \mid x, \theta_{jl}) = 1/[1+\exp(-x^T\theta_{jl})]$
- Overall Model for $Y$ is a mixture (latent class) model:
  $$
  \Pr(y \mid x, \Psi) = \sum_{j=1}^{K} g_j(x, \gamma_j) \sum_{\ell=1}^{K} g_{\ell|j}(x, \gamma_{j\ell}) \Pr(y \mid x, \theta_{j\ell})
  $$

where $\Psi$ is the collection of all parameters
Estimation

- Maximize log-likelihood of the data over all the parameters in $\Psi$.

- EM algorithm
  - E-step
    - Estimates latent class membership $\Delta$ for each $x$
  - M-step
    - Computed estimates are used as observation weights
Comparison with CART

- Similar to CART with linear combination splits
- Soft splits better able to capture gradual transitions from low to high response
- Very difficult to optimize tree topology of HME
- To date, emphasis has been on prediction rather than interpretation.
The End

Next Saga: Boosting and Additive Trees