Statistics 360: Advanced R for Data Science

MARS, part II

Becky Lin

Revisit: MARS model

- MARS model has the general form

$$f(x) = \beta_0 + \sum_{k=1}^{M} \beta_k h_k(x)$$

- $h_k(x)$ is a function from set C of candidate functions or a product of two or more such basis functions. Where

  Hinge function: $h(x) = \max(0, x)$

  Also note that Note that h(x-t) and h(t-x) are mirror images.

$$C = \{(x_j - t)_+, (t - x_j)_+\}_{t \in \{x_{1j}, x_{2j}, \ldots, x_{Nj}\}, j=1,2,\ldots,p}.$$

- βs are the coefficients estimated by minimizing the residual sum of squares (standard linear regression).
- These coefficients can be consider weights that represent the importance of the variable.

Revisit: MARS Model Building Procedure

1. Gather data: p input variables with n observations each, giving a total of np data points.
2. Calculate set of candidate functions by generating reflected pairs of basis functions with knows set at observed values.
3. Specify constraints; the number of terms in the model and maximum allowable degree of interaction.
4. **Do forward pass: try out new pair of basis function or function product and see which product decreases training error.**
5. Do backward pass: fix overfit.
6. Do generalized cross validation to estimate the optimal number of terms in the model.

Recursive partitioning

- In the week 4 exercises we will implement the forward part of an algorithm called recursive partitioning.
- Friedman (1991) recasts recursive partitioning in terms of a forward stepwise regression procedure that selects products of mirror-image step functions $I((x_v - t) \geq 0) = I(x_v \geq t)$ and $I(-(x_v - t) \geq 0) = I(x_v \leq t)$.
- Exercise: draw these two step functions to see why they are mirror image.
- Note: I think the first mirror-image step function should be $I(x_v > t)$, but will not make any changes to the discussion in the paper.

Region splitting as a product of step functions

- Take the case of two covariates and a region $R = [a_1, b_1] \times [a_2, b_2]$.
- Claim: R is the set of points (x_1, x_2) such that the basis function

$$B_R(x) = I(x_1 \geq a_1) \times I(x_1 \leq b_1) \times I(x_2 \geq a_2) \times I(x_2 \leq b_2) > 0$$

- Splitting R on variable x_v at point $t \in [a_v, b_v]$ means removing a "parent" basis function $B_R(x)$ from the model and replacing it with two "children" basis functions $B_R(x)I(x_v \leq t)$ and $B_R(x)I(x_v \geq t)$.

Recursive partitioning forward algorithm (paper page 11)

Algorithm 1 (recursive partitioning)

$B_1(\mathbf{x}) \leftarrow 1$
For $M = 2$ to $M_{\max}$ do: lof* $\leftarrow \infty$
  For $m = 1$ to $M - 1$ do:
    For $v = 1$ to n do:
      For $t \in \{x_{vj} | B_m(\mathbf{x}_j) > 0\}$
        $g \leftarrow \sum_{i \neq m} a_i B_i(\mathbf{x}) + a_m B_m(\mathbf{x}) H[+(x_v - t)] + a_M B_m(\mathbf{x}) H[-(x_v - t)]$
        lof $\leftarrow \min_{a_1, \ldots, a_M} \mathrm{LOF}(g)$
        if lof < lof*, then lof* $\leftarrow$ lof; $m^* \leftarrow m$; $v^* \leftarrow v$; $t^* \leftarrow t$ end if
      end for
    end for
  end for
  $B_M(\mathbf{x}) \leftarrow B_{m^*}(\mathbf{x}) H[-(x_{v^*} - t^*)]$
  $B_{m^*}(\mathbf{x}) \leftarrow B_{m^*}(\mathbf{x}) H[+(x_{v^*} - t^*)]$
end for
end algorithm

- Outer loop 1 over the number of model terms (M), from 1 to a specified maximum number of basis functions.
- Outer loop 2 over parent basis functions to replace by splitting
- Inner loops to choose variables v, splits t in the region where $B_R(x) > 0$, and coefficients for $B_R(x)I(x_v \leq t)$ and $B_R(x)I(x_v \geq t)$ to minimize a LOF criterion. (This is like our recursive partitioning.)

Week 6 lab (lab 4)

- Re-implement recursive partitioning with the forward stepwise regression algorithm outlined above.
- A more detailed breakdown of the tasks will follow.

MARS generalization

- Replace the step functions with hinge functions $h(t - x_v)$ and $h(x_v - t)$, where $h(x) = \max(0, x)$.
- Do not remove a parent basis function, just add pairs of children.
- Restrict the product that defines a basis function to distinct variables; i.e., no variable appears twice in the product.

Revisit: MARS Forward Pass

- At each step, MARS adds the basis function which reduces the residual error the most
- Always adds the basis function in "pairs", both sides of knot
- Calculate value for knot and function that fit the data, least squares.
- This is greedy algorithm.
- The addition of model terms continues until the max number of terms in the model is reached.

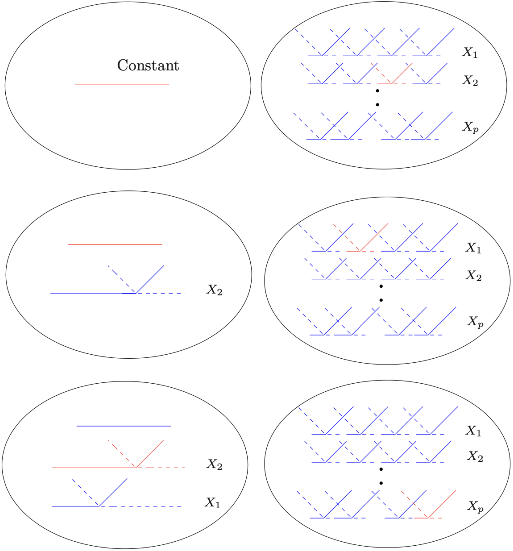

Figure 1: MARS forward pass (picture credit: ESL)

The first three steps in MARS forward model-building procedure.

MARS forward algorithm (MARS paper page 17)

Algorithm 2 (MARS—forward stepwise)

$B_1(\mathbf{x}) \leftarrow 1$; $M \leftarrow 2$
Loop until $M > M_{\max}$: $\text{lof}^* \leftarrow \infty$
  For $m = 1$ to $M - 1$ do:
    For $v \notin \{v(k, m) | 1 \le k \le K_m\}$
      For $t \in \{x_{vj} | B_m(\mathbf{x}_j) > 0\}$
        $g \leftarrow \sum_{i=1}^{M-1} a_i B_i(\mathbf{x}) + a_M B_m(\mathbf{x})[+(x_v - t)]_+ + a_{M+1} B_m(\mathbf{x})[-(x_v - t)]_+$
        $\text{lof} \leftarrow \min_{a_1, \ldots, a_{M+1}} \text{LOF}(g)$
        if $\text{lof} < \text{lof}^*$, then $\text{lof}^* \leftarrow \text{lof}$; $m^* \leftarrow m$; $v^* \leftarrow v$; $t^* \leftarrow t$ end if
      end for
    end for
  end for
  $B_M(\mathbf{x}) \leftarrow B_{m^*}(\mathbf{x})[+(x_{v^*} - t^*)]_+$
  $B_{M+1}(\mathbf{x}) \leftarrow B_{m^*}(\mathbf{x})[-(x_{v^*} - t^*)]_+$
  $M \leftarrow M + 2$
end loop
end algorithm

Figure 2: Algorithm 2-MARS forward

- Outer loop 1 over the number of model terms, M, from 1 to some maximum number
  - Outer loop 2 over parent basis functions B_m to generate children
    - Inner loops over variables v not part of B_m, splits t such that B_m is positive for $x_v = t$ and coefficients for the child basis functions to minimize a LOF criterion.