STAT361 Laboratory for Advanced R for Data Science

Lab 3

Sidi Wu

Feburary 16/17, 2023

Department of Statistics and Actuarial Science
Simon Fraser University

MARS - Algorithm 1

1. Step Function

$$H[\eta] = \begin{cases} 1 & \text{if } \eta \geq 0, \\ 0 & \text{otherwise,} \end{cases}$$

Previous version

```
Bnew <- data.frame(B[,(1:M)[-m]],
          Btem1=B[,m]*(x[,v]>t),Btem2=B[,m]*(x[,v]<=t))
gdat <- data.frame(y=y,Bnew)
lof <- LOF(y~.,gdat)
```

- The use of H(η) function is to split the parent basis function (B_m) into it's two children

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

Exercise:

1. Define function $H(\eta)$
2. Replace $(x[, v] > t)$ with $H(+(x_v - t))$
3. Replace $(x[, v] <= t)$ with $H(-(x_v - t))$

- $H(+(x_v - t))$: positive values will be indicated as 1
- $H(-(x_v - t))$: non-positive values will be indicated as 1

2. Record Splits (s, v, t) using Bfuncs

Each basis function B_m is a product of step functions

$$B_m(\mathbf{x}) = \prod_{k=1}^{K_m} H\left[s_{km} \cdot (x_{v(k,m)} - t_{km})\right]$$

- s: sign(+/-)
- v: index of the covariate
- t: split point
- K_m: number of splits in B_m
- m: index of the basis function

Bfuncs[[m]] is a data frame like below:

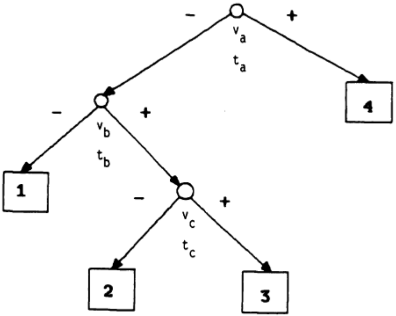
$$\begin{bmatrix} s_{1m} & v(1,m) & t_{1m} \\ s_{2m} & v(2,m) & t_{2m} \\ \cdots & \cdots & \cdots \\ s_{K_m m} & v(K_m,m) & t_{K_m m} \end{bmatrix}$$

Exercise:

1. Initialize **Bfuncs** to be an empty list of length $M_{\max}+1$ (you may use **vector()**)

Bfuncs

$$B_m(\mathbf{x}) = \prod_{k=1}^{K_m} H\left[s_{km} \cdot \left(x_{v(k,m)} - t_{km}\right)\right]$$



$$B_1 = H[-(x_{v_a} - t_a)]H[-(x_{v_b} - t_b)]$$

$$B_2 = H[-(x_{v_a} - t_a)]H[+(x_{v_b} - t_b)]H[-(x_{v_c} - t_c)]$$

$$B_3 = H[-(x_{v_a} - t_a)]H[+(x_{v_b} - t_b)]H[+(x_{v_c} - t_c)]$$

$$B_4 = H[+(x_{v_a} - t_a)]$$

FIG. 1. *A binary tree representing a recursive partitioning regression model with the associated basis functions.*

2. Record Splits (s, v, t) using Bfuncs

- **splits** will be replaced by **Bfuncs**
- Record the best split with a temporary object
- **Bfuncs** will be updated after a best split is given

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

Exercise (continued):

2. Copy the data frame **Bfuncs[[mstar]]** to **Bfuncs[[M+1]]** and add a row (s, v, t) to **Bfuncs[[M+1]]** with $s = -1$, and v, t from the best split
    - Add the splits of the new child basis function
3. Add a row (s, v, t) to **Bfuncs[[mstar]]** with $s = -1$, and v, t from the best split
    - Add the splits of the new child (sibling) basis function

3. Test the revised recpart_fwd()

Test your code as follows:

```
# Test
set.seed(123); n <- 10
x <- data.frame(x1=rnorm(n),x2=rnorm(n))
y <- rnorm(n)
rp_fwd <- recpart_fwd(y,x,Mmax=9)
rp_fwd$Bfuncs
```

Building R Package

Start an R package

Turn your skeleton implementation of MARS into an R package using the tools in the devtools package, as outlined in lecture 5.