$\mathcal{H} = \{ w_1 b_1(\vec{x}) + w_2 b_2(\vec{x}) + \dots + w_B \, b_B(\vec{x}) : \vec{w} \in \mathbb{R}^B \}$ $\quad y = \mathbb{R}$

and $b_1, b_2, \dots, b_B$ are known functions that attempt to span the function space of $f: \mathbb{R}^p \to \mathbb{R}$.

Example set of dimensions: set of all first-order interactions

$1, x_1, x_2, \dots, x_p, \; x_1^2, x_2^2, \dots, x_p^2, \; x_1 x_2, x_1 x_3, \dots, \; x_{p-1} x_p$ $B = 2p + \binom{p}{2}$

$\mid \mid \mid \mid \mid \mid \dots \mid \mid \mid \mid \dots, \; x_1^3, x_2^3, \dots, x_p^3, \; x_1^2 x_2, \dots x_{p-1}^2 x_p,$
$x_1 x_2 x_3, \dots, x_{p-2} x_{p-1} x_p$

Set of all second order interactions

  B is exponentially large $\{ x_2^{d_1} x_5^{d_2} x_6^{d_3} : i, j, k \text{ unique}, d_1 + d_2 + d_3 \le 3 \}$

A: likely... w's will be sparse, ie. most w's $= 0$. $d_n \in \{0, 1, 2, 3\}$

ⓞ Let $g(\vec{x}) = g_0(\vec{x}) = \bar{y}$, $X = [\vec{1}]$ ← OLS regression

① Try all B individually. $y \sim b_1(\vec{x}), \; y \sim b_2(\vec{x}), \dots, \; y \sim b_B(\vec{x})$ and compute SSE

  $X = [\vec{1} \; b_k(\vec{x})]$ reduction for each, $n \times 1$, keep best one b_{k_1}

② Try all $B - 1$ remaining individually $y \sim [\vec{1} \; b_{k_1}(\vec{x}) \; b_k(\vec{x})]$ and compute SSE
    reduction for each and keep best one b_{k_2}

⋮

Repeat

⑤ Stop if... ooSSE goes up.

Ⓓ $\boxed{\mathcal{D}_{train}}$
    $\boxed{\mathcal{D}_{sel}}$
    $\boxed{\mathcal{D}_{test}}$

Model Selection Procedure

$g_1, g_2, \dots, g_m$ $m = 2^B$

The algorithm could be modified to use K-fold CV (more & more).

Classification and Regression Tree (CART) Algorithm (1984).

  $y = \mathbb{R}$ $A: OLS \Rightarrow g(x) = \bar{y}$

bin size $= 1$

$\mathcal{H}_{B_1} = \{ w_1 \mathbb{1}_{x \in [0,1)} + w_2 \mathbb{1}_{x \in [1,2)} + \dots + w_9 \mathbb{1}_{x \in [9, 10]} : \vec{w} \in \mathbb{R}^9 \}$

$A: OLS \Rightarrow g(x) =$

$\mathcal{H}_{B_{0.1}} = \{ w_1 \mathbb{1}_{x \in [0, 0.1]} + \dots + w_{99} \mathbb{1}_{x \in [9.1, 10]} : \vec{w} \in \mathbb{R}^{99} \}$

just right

$\mathcal{H}_{B_{3.3}} = \{ w_1 \mathbb{1}_{x \in [0, 3.3)} + w_2 \mathbb{1}_{x \in [3.3, 6.6)} + w_3 \mathbb{1}_{x \in [6.6, 10]} : \vec{w} \in \mathbb{R}^3 \}$

Select bin size using the ooSSE curve

In two dimensions $(p=2)$, the bins are squares. $B = \#\text{bins}/\text{dim} \Rightarrow B_{tot} = B^2$

In p dimensions $B_{tot} = B^p$ which $\gg n$ very fast.

A solution: "binary trees" with split rules that are orthogonal to an axis.

$= $ union of mutually exclusive, collectively exhaustive, possibly infinitely large hyperranges.

Regression Tree Algorithm

ⓞ Let distances be all done   ordered/sorted values

① Consider every possible orthogonal-to-axis split $x_j \le t$, $j = 1, \dots, p, \; c \in 1, \dots, n-1$ and compute SSE_L, SSE_R the SSE's in the putative left node and right node. Select the rule where

$$SSE_{weighted} := \frac{n_L \, SSE_L + n_R \, SSE_R}{n_L + n_R} \qquad n_L := \# \text{ observations in left set}$$
$$n_R := \text{''} \quad \text{'' '' '' right ''}$$

is smaller, ie. create an inner node with this split rule and a left leaf with $\hat{y} = \bar{y}_L$ and a right leaf with $\hat{y} = \bar{y}_R$

② If $n_L \ge N_0$ then set distances $=$ left partition and recursively do it.
   If $n_L > N_0$ '' '' '' '' '' right ''

N_0 is a hyperparameter. If N_0 is small e.g. $N_0 = 1 \Rightarrow g$ is overfit.
If N_0 large $\Rightarrow$ underfit model. How to pick N_0? Use 3-fold selection.