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

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 functions is set of all first-order interactions

$1, x_1, x_2, \dots, x_p, x_1^2, x_2^2, \dots, x_p^2, x_1x_2, x_1x_3, \dots, x_{p-1}x_p \qquad B = 2p + \binom{p}{2}$

set of all second order interactions

B is exponentially large

$1, \dots, x_1^3, x_2^3, \dots, x_p^3, x_1^2x_2, \dots x_{p-1}^2 x_p, x_1x_2x_3, \dots, x_{p-2}x_{p-1}x_p$

$\{ x_i^{d_1} x_j^{d_2} x_k^{d_3} : i,j,k \text{ unique}, d_1+d_2+d_3 \le 3, d_m \in \{0,1,2,3\} \}$

A: likely... w's will be sparse i.e. most w's $= 0$.

② 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

$X = [\vec{1}\ b_k(\vec{x})]$

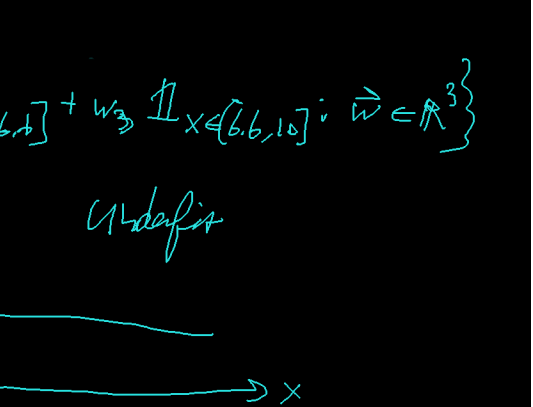
② Try all $B-1$ remaining individually $y \sim [\vec{1}\ b_{k_1}(\vec{x})\ b_k(\vec{x})]$ and compute SSE

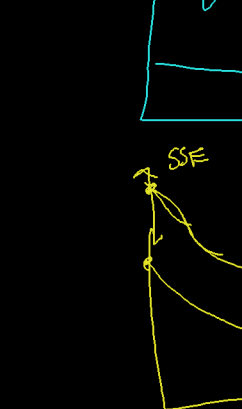
Keep best one b_{k_1}

Keep best one b_{k_2}

Repeat

⑤ Stop if... OOSSE goes up.





Model Selection Problem

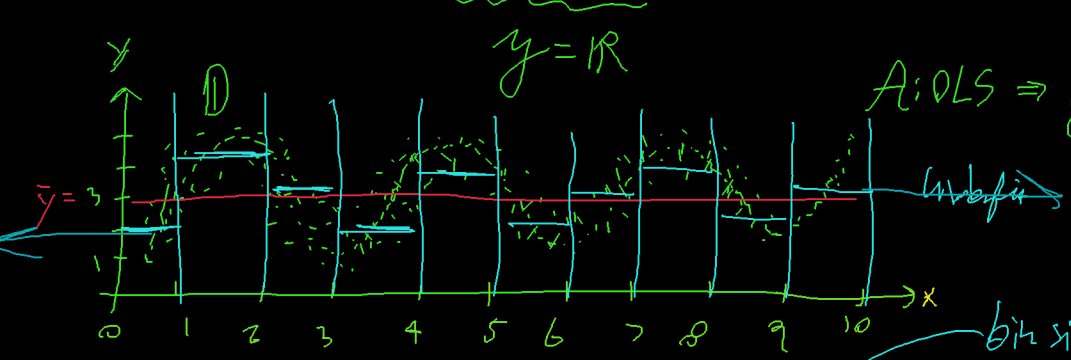
$g_1, g_2, \dots, g_M \qquad M = 2^B$

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

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

$y = \mathbb{R}$

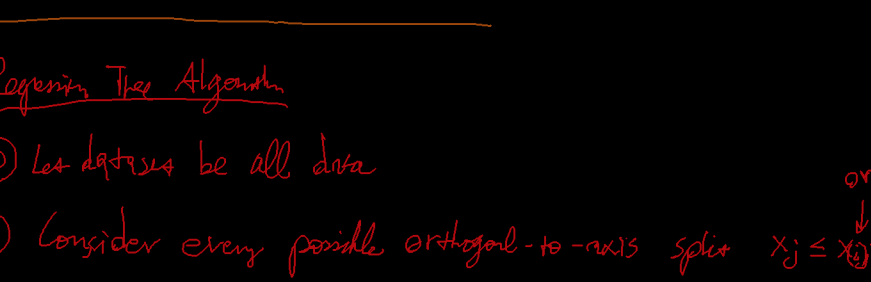
A: OLS $\Rightarrow g(x) = \bar{y}$



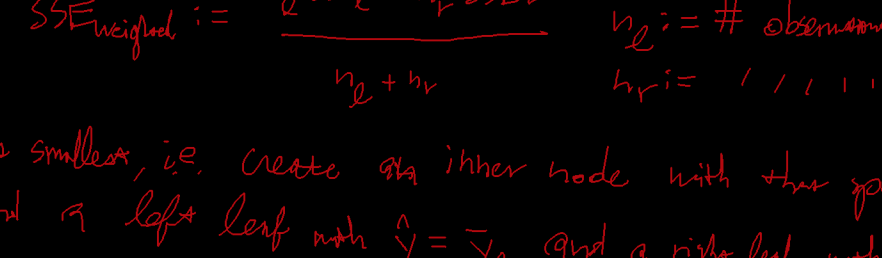
$\mathcal{H}_{B1} = \left\{ w_1 \mathbb{1}_{x \in [0,1]} + w_2 \mathbb{1}_{x \in (1,2]} + \dots + w_9 \mathbb{1}_{x \in [9,10]} : \vec{w} \in \mathbb{R}^9 \right\}$

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

$\mathcal{H}_{B0.1} = \left\{ w_1 \mathbb{1}_{x \in [0,0.1]} + \dots + w_{99} \mathbb{1}_{x \in [9.9,10]} : \vec{w} \in \mathbb{R}^{99} \right\}$



$\mathcal{H}_{B3.3} = \left\{ w_1 \mathbb{1}_{x \in [0,3.3]} + w_2 \mathbb{1}_{x \in (3.3,6.6]} + w_3 \mathbb{1}_{x \in (6.6,10]} : \vec{w} \in \mathbb{R}^3 \right\}$



Select bin size using the OOSSE curve

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

In p dimensions $B_{tot} = B^p$ which $>$ very fast.

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

= Union of mutually exclusive, collectively exhaustive, possibly infinitely large hyperrectangles.

Regression Tree Algorithm

⓪ Let dataset be all data

① Consider every possible orthogonal-to-axis split $x_j \le x_{(i)j}$ ← ordered / sorted values $j = 1 \dots p, i \in 1, \dots, n-1$.

and compute SSE_ℓ, SSE_r the SSE's in the potential left node and right node. Select the rule where

$$SSE_{weighted} := \frac{n_\ell SSE_\ell + n_r SSE_r}{n_\ell + n_r}$$

$n_\ell :=$ # observations in left set

$n_r :=$ " " " right "

is smallest, i.e. create an inner node with that split rule and a left leaf with $\hat{y} = \bar{y}_\ell$ and a right leaf with $\hat{y} = \bar{y}_r$.

② If $n_\ell > N_0$ then set dataset = left partition and run step 1 on it.

If $n_r > N_0$ " " " right " " " " " " .

Also: don't split if all the y's are the same

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.