Lecture 18:

$H = \{ w_1 b_1(\vec{X}) + w_2 b_2(\vec{X}) + \dots + w_B b_B(\vec{X}) ; \vec{w} \in \mathbb{R}^B \}$

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

Example set of functions :- set of all first order interaction

$X_1, X_2, \dots, X_p, X_1^2, X_2^2, \dots, X_p^2 ; X_1 X_2, X_1 X_3, \dots, X_{p-1} X_p$

$\dots, X_1^3, X_2^3, \dots, X_p^3,$

$X_1^2 X_2, \dots, X_{p-1}^2 X_p,$

$X_1 X_2 X_3, \dots, X_{p-2} X_{p-1} X_p.$

$\boxed{B = 2p + \binom{p}{2} = 2^p}$

Set of all second interactions

B is exponentially large.

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

Forward Stepwise OLS

OLS representation

⓪ Let $g(\vec{X}) = g_0(\vec{X}) = \bar{Y}$

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 reduction for each.

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

② Try all B−1 remaining individually

$y \sim [\vec{1}\ b_{*1}(\vec{x})\ b_k(\vec{x})]$ and compute

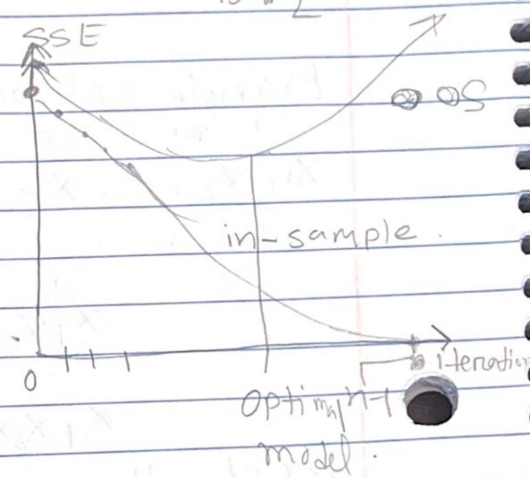
SSE reduction for each and best one b_{*2}

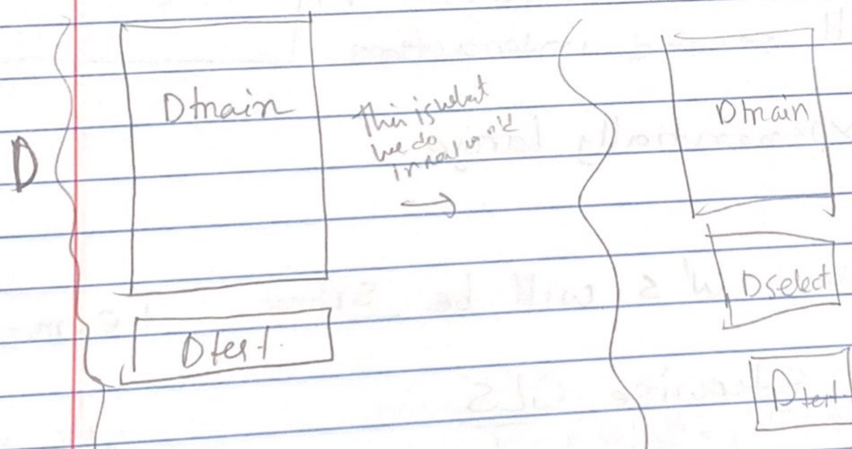
⋮

repeat

⋮

③ Stop if ... (outer sample) OOS SSE goes up.





Model selection Procedure

$g_1, g_2, \ldots, g_m$

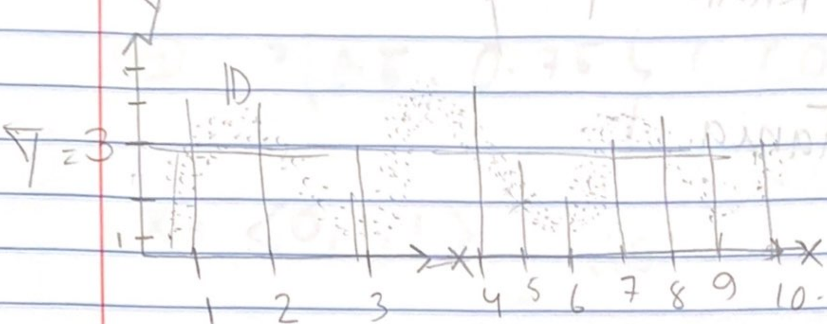
$m = 2^B$

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

$y = \mathbb{R}$

Classification and regression Tree Cart Algorithm (1984)

A : OLS $\Rightarrow g(X) = \bar{Y}$

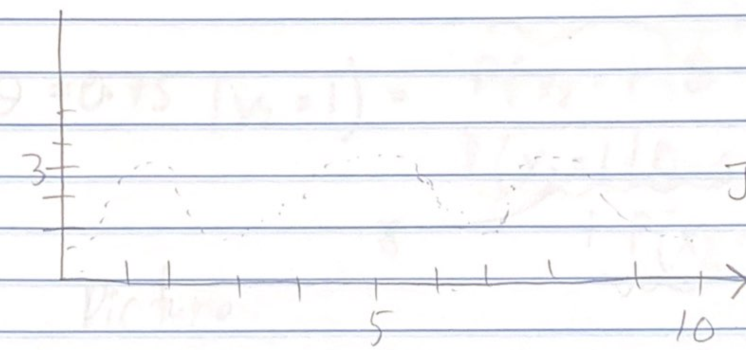


Underfit

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

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

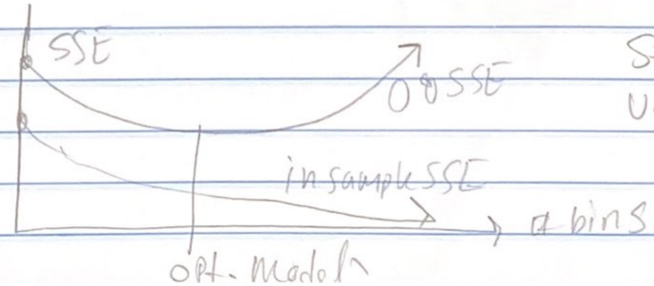
$\mathcal{H}_{B0.1} = \{ w_1 \mathbb{1}_{x \in [0,.1]} + w_{99} \mathbb{1}_{x \in [9.9,10]} : \vec{w} \in \mathbb{R}^{99} \}$



Just right.

$\mathcal{H}_{B3.3} = \{ w_1 \mathbb{1}_{x \in [0,3.3]} + w_2 \mathbb{1}_{x \in [3.3,6.6]}$

Underfit.



Select bin size using the OOSSE curve

Next notebook..

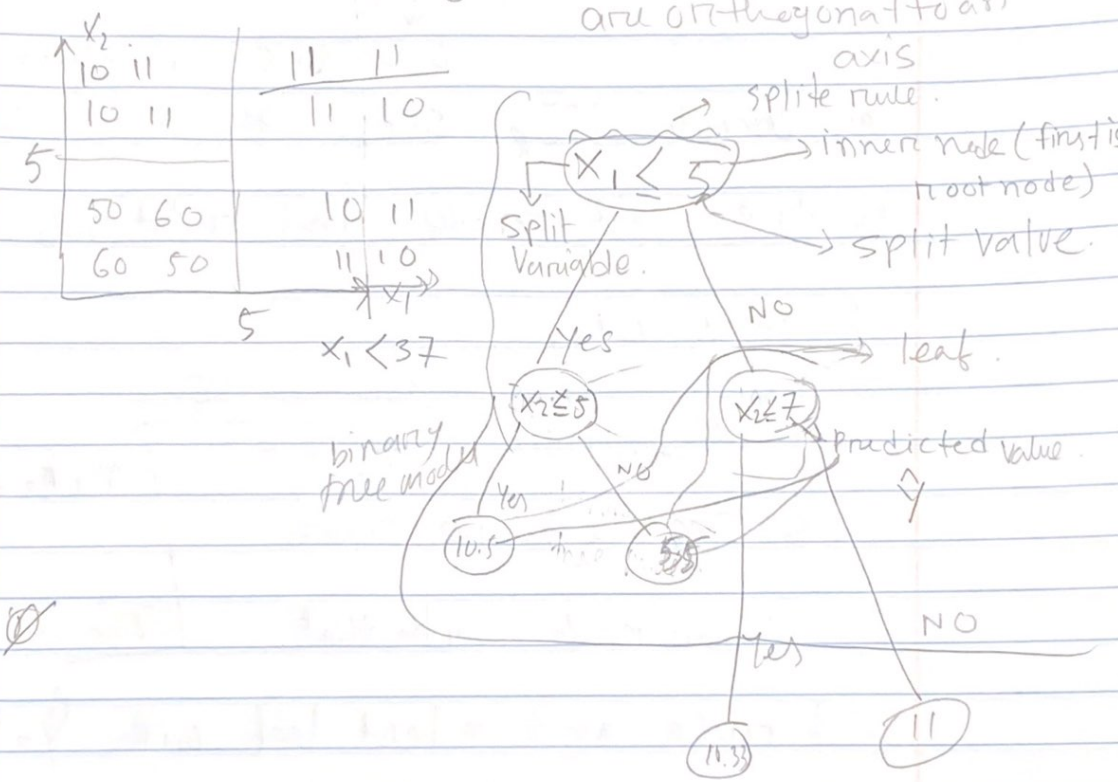
~~Leafs~~

In two dimensions $(P=2)$, the bins are squares

$B = \#$ bins/dim $\Rightarrow B_{tot} = B^2$

In P dimensions $B_{tot} = B^P$ which $> n$ very fast

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



X

= union of mutually exclusive,

Collectively exclusive, Possibly infinitely

large hyper______

Regression Tree algorithm;

(0) let dataset be all data

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

ordered / sorted values.

and compute SSE_{ℓ}, SSE_{r} the SSE's in the putative left node and right node. Select the rule where,

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

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

$n_r :=$ # " right "

is smallest. ie create 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 and run step 1

if $n_r > N_0$ " " = right " "

N_0 is a hyperparameter. If N_0 is small e.g

$N_0 = 1 \Rightarrow g$ is overfit

If N_0 is large $\Rightarrow$ Underfit model How to pick

N_0? Use 3-fold selection.