Math 390 4/12/19 Lec 20

Bin sizes selected by model selection procedure.

problem: multiple dimensions has lots of bin sizes. If B bins per feature, total # bins B^p i.e. increase exponentially $B^p > n$ almost instantly → still overfitting. Need other way.

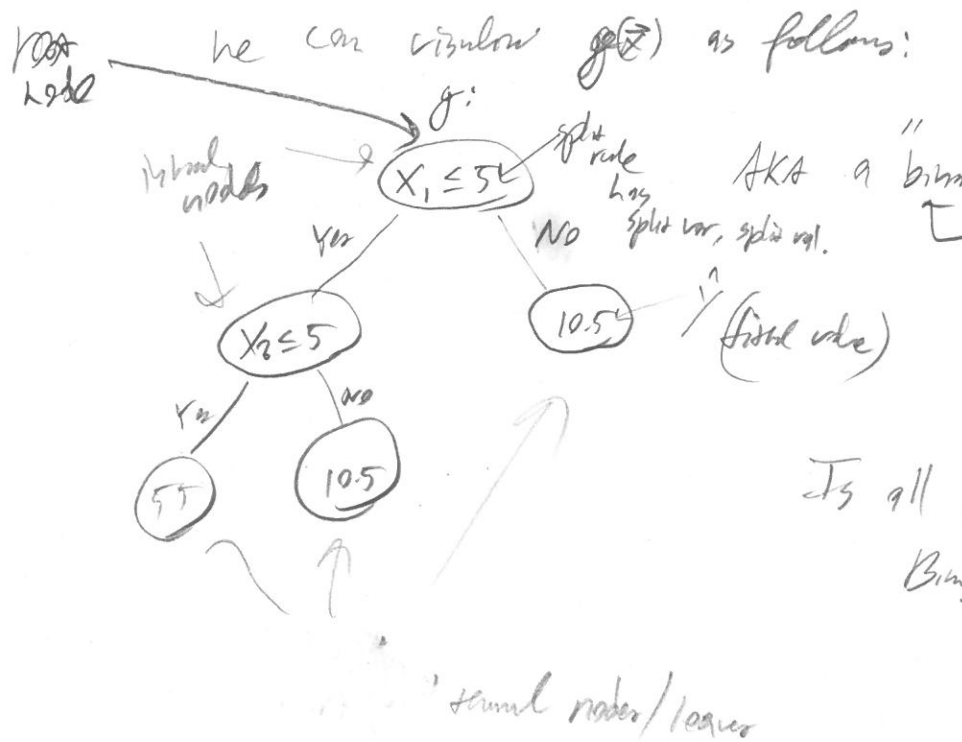
Consider the following data with $p=2$

A good model would be the following

$$g(\vec{x}) = 10.5\,\mathbb{1}_{x_1>5} + 10.5\,\mathbb{1}_{x_1\le 5}\mathbb{1}_{x_2>5} + 55\,\mathbb{1}_{x_1\le 5}\mathbb{1}_{x_2\le 5}$$

this is a linear model with derived features!

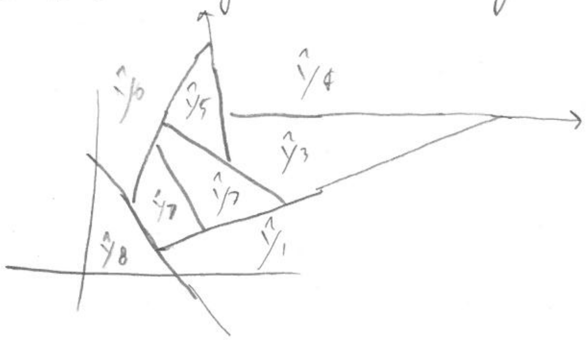
We can visualize $g(\vec{x})$ as follows:



AKA a "binary tree" ← binary choices at each internal node

Is all $\mathcal{X}$ acct for? Yes! Binary splits of the form $x_j \le v$ guarantee this

Binary tree models with splits of this form divide $\mathcal{X}$ into hyperrectangles that are "parallel to the axes" e.g. this is not possible:



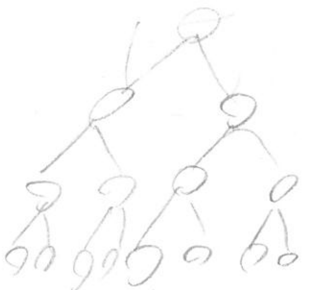
These hyperrectangles are the results of an arbitrary ~~featurization~~ transformation of the original p raw features. What $\mathcal{A}$ can do all this? Consider this one.

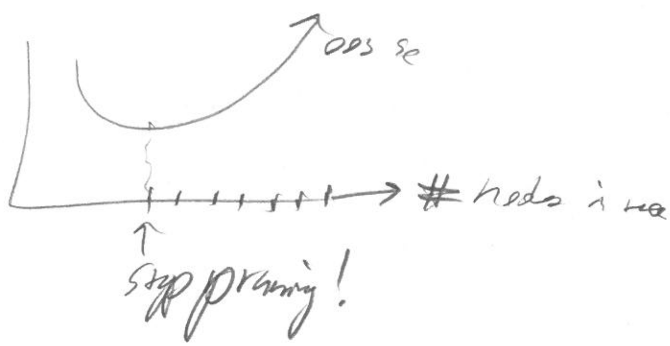
Regression Tree

1. Begin with all data $\langle X, y \rangle$
2. Consider every possible parallel-to-axes split $\langle X_\ell, y_\ell \rangle, \langle X_r, y_r \rangle$ and rules
   $x_1 \le x_{(1)1}, x_1 \le x_{(2)1}, \ldots, x_1 \le x_{(n-1)1}$, $x_2 \le x_{(1)2}, \ldots, x_2 \le x_{(n-1)2}, \ldots, \ldots$
   sort $\langle X, y \rangle$ by $\vec{x}_{\cdot 1}$ $\quad$ sort $\langle X, y \rangle$ by $\vec{x}_{\cdot 2}$
   and calculate costs: $SSE_\ell := \sum (y_\ell - \bar{y}_\ell)^2$, $SSE_r := \sum (y_r - \bar{y}_r)^2$
3. Locate rule with smallest weighted avg:
   $$\frac{n_\ell SSE_\ell + n_r SSE_r}{n_\ell + n_r}$$
4. Create the split with that rule and assign $\hat{y} = \bar{y}$ to each node.
5. Recursively run steps 2-4 on each daughter node, stop if node only has N_0 obs's inside.

If N_0 large $\to$ underfitting. If N_0 low $\to$ overfitting. If $N_0 = 1$, each obs has its own node which means $\hat{y} = y$ and you get exact overfitting.

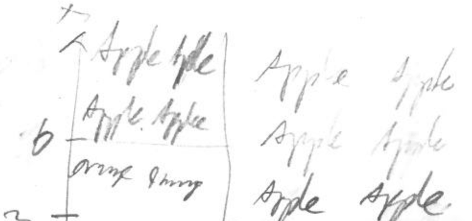
How to prevent overfitting? Let $N_0 = 1$ and then "prune". Collapse tree back to internal node one-by-one by looking at marginal reduction in OOS s_e. Continue until s_e starts to ~~go~~ go up





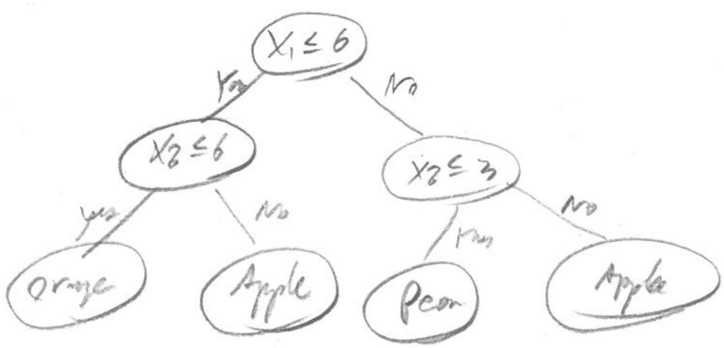
Kind of like backwards stepwise regression. $\boxed{\text{DEMO}}$

If $\mathcal{Y} = \{1, 2, \ldots, L\}$ multilabel classification. If $L=2$, binary classification.



If Binary classification, we discussed perceptron, SVM with logistic objective. Both are linear models and thus suffer from some problem of model selection. Also they can't build models like so

Tree:



Written as a linear model:

$$\hat{y} = g(x) = \text{Orange}\, \mathbb{1}_{x_1 \le 6} \mathbb{1}_{x_2 \le 6} + \text{Apple}\, \mathbb{1}_{x_1 \le 6} \mathbb{1}_{x_2 > 6} + \text{Pear}\, \mathbb{1}_{x_1 > 6} \mathbb{1}_{x_2 \le 3} + \text{Apple}\, \mathbb{1}_{x_1 > 6} \mathbb{1}_{x_2 > 3}$$

Algorithm to generate this: Same thing as the regression algorithm except ① in step 2, the cost becomes

$$\text{Gini} := \sum_{\ell=1}^{L} \hat{p}_\ell (1 - \hat{p}_\ell) \quad \text{where} \quad \hat{p}_\ell := \frac{\#\, y_i \text{ in level } \ell}{\#\text{ in node}}$$