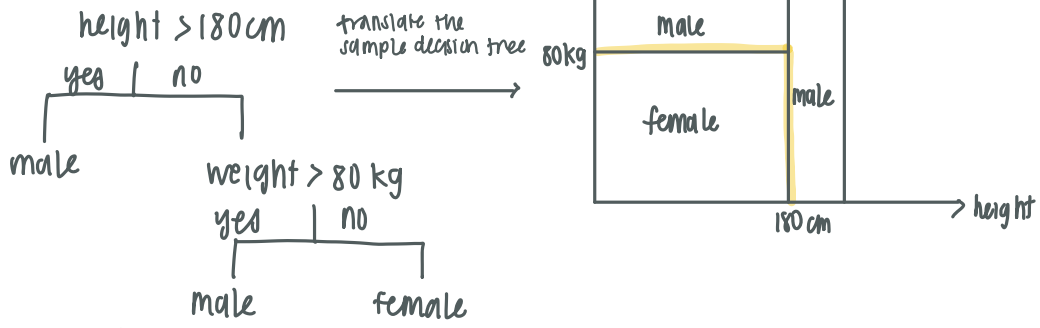
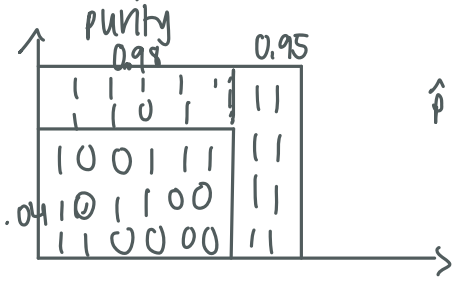
Trees

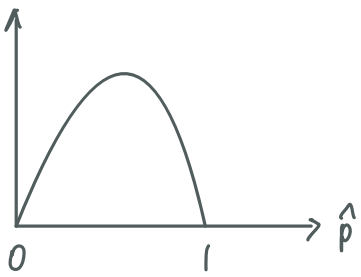
classification decision tree





$\hat{p}$ = prob(male) for a region

gini: $\hat{p}(1-\hat{p})$ (x samples in the region)



each partition leads to maximal improvement in overall purity

gini index: weight each partitioning by n_{region} to observe reduction

each iteration:

choose a region

choose a variable to partition on

choose a cut point

} exhaustive search to determine optimal partitioning for recursive partition

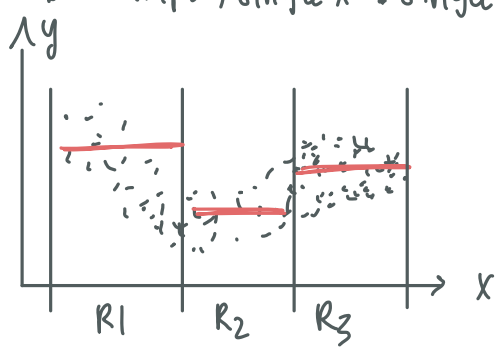
a heuristic search

↘ max reduction in $Loss(R_1, \dots, R_m \dots R_p)$

stop if reduction $< \lambda$.

Regression trees

1-D example, single X & single y



$y = f(X) + \epsilon$

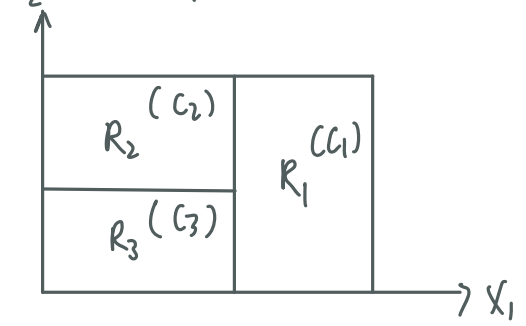
$f(X) = c_1 \mathbf{1}(X \in R_1) + c_2 \mathbf{1}(X \in R_2) + c_3 \mathbf{1}(X \in R_3)$

↓

$= \begin{cases} 1 \text{ if } X \in R_1 \\ 0 \text{ if } X \notin R_1 \end{cases}$

Regression trees (continued)

X_2 2-D example



Least squares loss

$$\text{Loss} = \sum_{i=1}^{n} (y_i - f(x_i))^2$$

piecewise approximation

$$= \sum_{m=1}^{M} \sum_{x_i \in R_m} (y_i - c_m)^2 + \lambda M$$

add penalty to grow shadow tree

$$\text{Loss}(R_1 \dots R_m, \dots R_M)$$

$$\frac{\partial}{\partial c_m} = \sum_{x_i \in R_m} (y_i - c_m) = 0$$

$$\sum_{x_i \in R_m} y_i - N_m$$

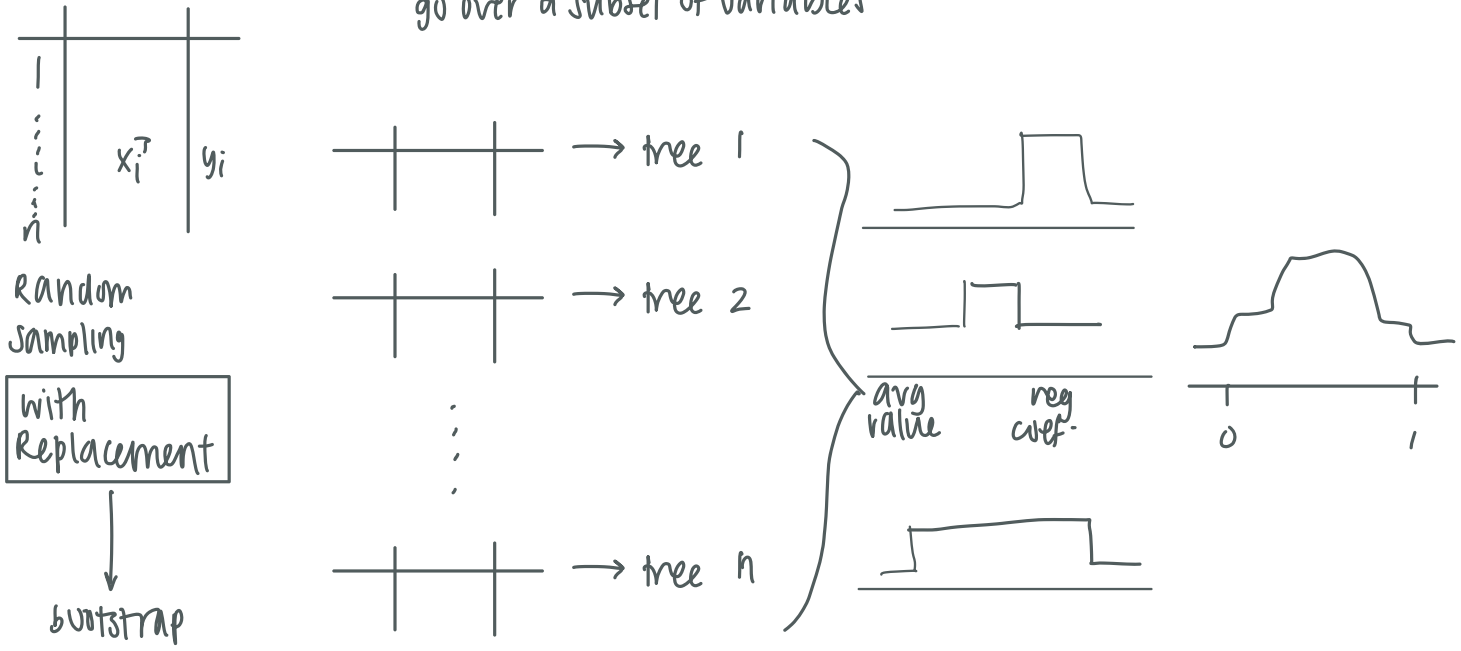
↓

number of $x_i \in R_m$

$$\hat{c}_m = \frac{\sum_{x_i \in R_m} y_i}{N_m} = \frac{\sum_{i=1}^{n} y_i \, \mathbb{1}(x_i \in R_m)}{\sum_{i=1}^{n} \mathbb{1}(x_i - R_m)}$$

Forest

each split
go over a subset of variables

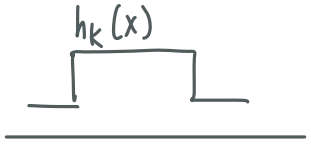


boosting

$$f(x) = \sum_{k=1}^{K} h_k(x)$$

ensemble machine
committee machine
add trees sequentially

$h_k(x)$ is a tree



$$\text{Loss} = \sum_{i=1}^{n} (y_i - f(x_i))^2$$

current committee

$$s_i = \boxed{\sum_{k=1}^{t-1} h_k(x_i)}_{\hat{s}_i} + \boxed{h_t(x_i)}$$

current committee (fixed)

new member (tree) to be grown

$$\text{Loss} = \sum_{i=1}^{n} (y_i - s_i)^2$$

$$= \sum_{i=1}^{n} (y_i - (\hat{s}_i + h_t(x_i))^2$$

$$= \sum_{i=1}^{n} (\underbrace{\boxed{y_i - \hat{s}_i}}_{\hat{e}_i} - h_t(x_i))^2$$

$$= \sum_{i=1}^{n} (\hat{e}_i - h_t(x_i))^2$$

treat $\hat{e}_i$ as response

small jumps:

we penalize c_m

$$\text{Loss} = \sum_{m=1}^{M} \sum_{x_i \in R_m} (\hat{e}_i - c_m)^2 + \lambda M + \sum_{m=1}^{M} \gamma c_m^2$$

gamma

Logistic regression

Iterated reweighted least squares

$$\sum_{i=1}^{n} \hat{w}_i \left(\hat{y}_i - h_t(x_i)\right)^2$$

↑ ("$x_i^T \Delta\beta$ substitute")

$$\frac{\hat{e}_i}{\hat{w}_i} \leftarrow y_i - \hat{p}_i \leftarrow \frac{e^{\hat{s}_i}}{1+e^{\hat{s}_i}}$$

$$\hat{w}_i \leftarrow \hat{p}_i(1-\hat{p}_i)$$

extreme gradient boosting

Instead of finding $\Delta\beta$, we grow a new tree to expand the error

each tree outputs a continuous score & does not withhold info compared to classification trees (binary)

Adaboost

less commonly used than extreme gradient boosting

$$s_i = f(x_i) = \sum_{k=1}^{K} \beta_k \boxed{h_k(x_i)}$$

s_i: strong classifier

$\boxed{h_k(x_i)}$: weak classifier

↓ classification tree $\rightarrow \{+1, -1\}$

$$\hat{y}_i = \text{sign}(s_i)$$

each step requires to grow a new tree + calculate $\hat{\beta}$