Impurity

- Gini impurity ≈ how often will random labels disagree
- Entropy ≈ information
- Least squares

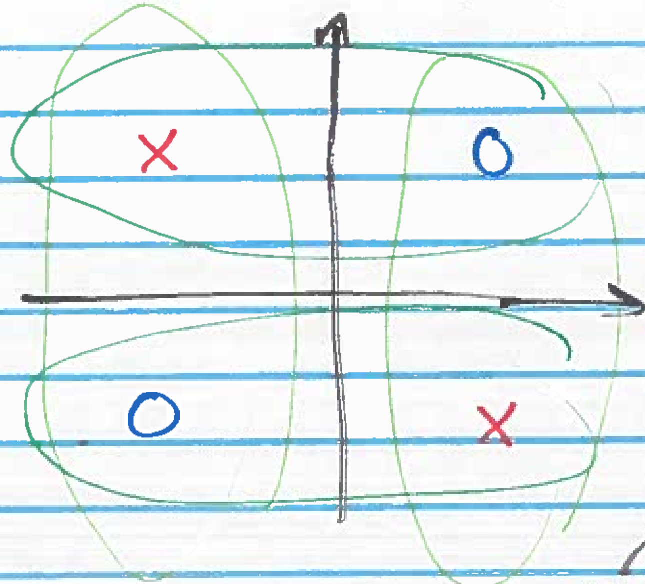
ID3

Input: Dataset $\mathcal{D}$
Search all features:
  search all splits for feature:
    evaluate impurity (entropy)
pick feature/split that minimizes impurity
construct node
recursively call ID3 on subsets

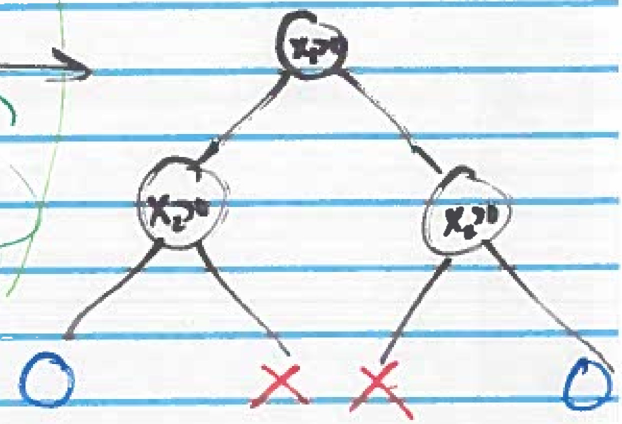
How to split?
- categorical features → 1 child per category split
- real-valued features → threshold

Base cases: if $|D| = 1$ — classification → majority; regression → average
1. $\exists$ exists some $\hat{x}$ s.t. $\forall (x, y) \in \mathcal{D}$, $x = \hat{x}$
2. exists $\hat{y}$ s.t. $\forall (x, y) \in \mathcal{D}$, $y = \hat{y}$ ⇒ leaf $\hat{y}$
3. $|D| = 0$, predict majority/average of "parent" dataset

Why not stop when impurity doesn't decrease.



$p_x = \frac{1}{2}$ $p_o = \frac{1}{2}$



Decision tree inference time is proportional to depth

$\Downarrow$

inference very fast!

Overfitting:

$\Uparrow$

$\text{bias}^2 + \text{variance} + \text{noise}$

Ensembling: average the prediction of some models

- draw m independent datasets $D_1, \ldots D_m$
- for each dataset:
  - run ID3 $\rightarrow$ hypothesis h_i
- output $\hat{h}(x) = \frac{1}{m} \sum_{i=1}^{m} h_i(x)$

How does this effect: bias^2 variance noise ?

same $\downarrow$ factor of m same

$$Var\left(\frac{1}{m}\sum_{i=1}^{m} h_i(x)\right) = \frac{1}{m^2} Var\left(\sum_{i=1}^{m} h_i(x)\right)$$

$$= \frac{1}{m^2} \sum_{i=1}^{m} Var(h_i(x))$$

$$= \frac{1}{m^2} \cdot m \cdot Var(h_i(x))$$

$$= \frac{1}{m} Var(h_i(x))$$

Bootstrap Aggregating

instead of drawing from source dist,
we draw with replacement from D

given D ~~subsampled~~
~~draw m ~~ ~~dataset~~
draw m datasets from D each of size n
for each, train a decision tree (ID3)
average: $\hat{h}(x) = \frac{1}{m}\sum_{i=1}^{m} h_i(x)$.

still reduces variance!
- idea: individual examples are still distributed according to P (source dist)
- even though they're not independent, they're "independent enough" to reduce variance.

Random forest ⇒ full algorithm Bagging + Trees