Impurity

- Gini impurity \( \approx \) how often will random labels disagree
- Entropy \( \approx \) information
- Least squares

**ID3**

Input: Dataset \( D \)

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 \( \rightarrow \) 1 child per category
- real-valued feature \( \rightarrow \) threshold

**Base cases:** if \( |D| = 1 \)

1. \( \exists \) exists, some \( \hat{x} \) s.t. \( \forall (x,y) \in D, \ x = \hat{x} \)
2. \( \exists \) exists \( \hat{y} \) s.t. \( \forall (x,y) \in D, \ y = \hat{y} \) \( \rightarrow \) leaf \( \hat{y} \)
3. \( |D| = 0 \), predict majority/average of "parent" dataset
Why not stop when impurity doesn't decrease.

\[ P_x = \frac{1}{2} \quad P_0 = \frac{1}{2} \]

Decision tree inference time is proportional to depth.

Inference very fast!

Overfitting:

\[ \text{bias}^2 + \text{variance} + \text{noise} \]
Ensembling: average the prediction of some models

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

How does this effect: bias? variance? noise?

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

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

\[
= \frac{1}{m} \cdot m \cdot \text{Var} (h_i(x))
\]

\[
= \frac{1}{m} \text{Var} (h_i(x))
\]
Bootstrap Aggregating

Instead of drawing from source dist, we draw with replacement from $D$

given $D$ sampled

draw $n$ datasets

draw $n$ datasets from $D$ each of size $N$

for each, I train a decision tree ($T_0$)

average: $\hat{h}(x) = \frac{1}{m} \sum_{i=1}^{m} h_i(x)$

still reduces variance!

- idea: individual examples are still i.i.d. dist. distributed according to $P$

- even though they're not independent, they're "independent enough" to reduce variance.

Random Forest $\Rightarrow$ full algorithm Baggng+Trees