Tree models

- **A Decision Tree** is a tree contains *a root, internal nodes, leaf nodes*, connected by *directed edges*.
  - Each *non-leaf node* **partitions** the data by some *features*.
- e.g. For linear models $f(x) = \text{sign}(w^T x + b)$
  - Decision logic:
    - if $w^T x + b \geq 0 \quad \Rightarrow \quad +1$
    - if $w^T x + b < 0 \quad \Rightarrow \quad -1$
- e.g. $y \in \{-1, +1\}$ represents a bad/good scientist
  - $x^{(1)} = 1$: is hard-working
  - $x^{(2)} = 1$: has a good vision
  - $x^{(3)} = 1$: likes banana
- How to select a good feature for partition?
  - **Purity**: 简单来说就是越早分出叶节点越好，这样决策树高度会更小
  - from Purity to **Entropy**

Entropy

- For a random variable X, entropy $H(x)$ is:

$$\begin{aligned} H(x) &= \sum_X p(x) \log \frac{1}{p(x)} \\ &= -\sum_X p(x) \log p(x) \\ &= E_X[\log \frac{1}{p(x)}] \end{aligned}$$

  - For an event $X = x$, the smaller $p(x)$ is, the more "information" it contains
    - $x :=$ "sun rises from the east", then $p(x) = 1$, contains zero information
    - $x :=$ "roll a dice 3 times, get 666", then $p(x) = \frac{1}{6^3}$, contains a lot of information
  - $\log \frac{1}{p(x)}$ measures the "information" of x
    - here $\log x = \log_2 x$, not $\ln x$
  - Entropy is **the expectation of "information"** of all events in a system.
  - $H(x) \geq 0$, takes "=" if and only if $\exists x,\ p(x) = 1$
  - $H(x) = \sum_X p(x) \log \frac{1}{p(x)} \leq \log \sum_X p(x) \frac{1}{p(x)} = \log n$, takes "=" if and only if $\forall x,\ p(x) = \frac{1}{n}$
    - n is $\#$ of events
    - log function is convex, so we can use Jensen's inequality
    - when all events in a system have **equal probability**, entropy $H(x)$ maximizes
- $H(x)$ measures **the disorder, randomness, uncertainty** of a system.
- Cross Entropy $H(p, q) = -\sum_X p(x) \log q(x)$
  - $q(y_i = 1) = \sigma(w^T x_i + b)$
  - $q(y_i = 0) = 1 - \sigma(w^T x_i + b)$
  - $p(y_i)$ measures the observed data, is a data distribution

$$\begin{aligned} H(x_i, y_i) &= -\sum_{y_i \in \{0,1\}} p(y_i) \log q(y_i) \\ &= -\big[y_i \log(\sigma(w^T x_i + b)) + (1 - y_i) \log(1 - \sigma(w^T x_i + b))\big] \end{aligned}$$

- We have 4 entropy scores can measure the entropy of a decision tree.

Information Gain

- $g(D, A) := H(D) - H(D|A)$
  - D is training set
  - A is a feature/attribute. $A \in \{a_1, \ldots, a_m\}$ has m discrete values.
    - If A is continuous
      - Sort training data's A values: $\alpha_1, \ldots, \alpha_n$
      - Take $\frac{\alpha_i + \alpha_{i+1}}{2}$ as $n - 1$ thresholds

            - Transform a continuous A into discrete
    - Suppose $y \in \{1, 2, \dots, K\}$
    - $H(D)$

$$H(D) := -\sum_{k=1}^{K} \frac{|C_k|}{|D|} \log \frac{|C_k|}{|D|}$$

        - $|C_k|$ is the set of training data whose $y = k$
        - $\frac{|C_k|}{|D|}$ approximates $P(y = k)$
        - $H(D)$ approximates $-\sum_{k=1}^{K} P(y = k) \log P(y = k) = H(y)$
    - $H(D|A)$

$$\begin{aligned} H(D|A) :&= \sum_{i=1}^{m} \frac{|D_i|}{|D|} \cdot H(D|A = a_i) \\ &= \sum_{i=1}^{m} \frac{|D_i|}{|D|} \big[-\sum_{k=1}^{K} \frac{|D_i \bigcap C_k|}{|D_i|} \log \frac{|D_i \bigcap C_k|}{|D_i|} \big] \end{aligned}$$

        - D_i is the set of training data whose feature $A = a_i$
        - $\frac{|D_i \bigcap C_k|}{|D_i|}$ approximates $P(y = k|A = a_i)$

$$\begin{aligned} H(D|A = a_i) &\approx -\sum_{k=1}^{K} P(y = k|A = a_i) \log P(y = k|A = a_i) \\ &= H(y|A = a_i) \end{aligned}$$

    - $H(D|A) \searrow \quad \Rightarrow \quad g(D, A) \nearrow \quad \Rightarrow \quad \text{Purity} \nearrow$
        - Select A that maximizes $g(D, A)$
- e.g. Suppose 2 features A, B, A has 2 discrete values with same probability and B has 10 discrete values with same probability. Class $y \in \{1, \dots, 10\}$ is pure in each $B = b_i$, that is, $P(y = i|B = b_i) = 1$.
  $P(y = i|A = a_1) = \frac{1}{5}, \forall i = 1, \dots, 5$
  $P(y = i|A = a_2) = \frac{1}{5}, \forall i = 6, \dots, 10$
    - $H(D) = \log 10$
    - $H(D|A) = \log 5 \quad H(D|B) = 0$
    - $g(D, A) = 1 \quad g(D, B) = \log 10$
    - B has better purity but may have lower generalization ability.

Information Gain Ratio

- $g_R(D, A)$

$$g_R(D, A) = \frac{g(D, A)}{H_A(D)}$$

    - $H_A(D)$

$$H_A(D) := -\sum_{i=1}^{m} \frac{|D_i|}{|D|} \log \frac{|D_i|}{|D|}$$

        - $\frac{|D_i|}{|D|}$ approximates $P(A = a_i)$
        - not entropy of y but A
        - When each $A = a_i$ has equal probability, $H_A(D) = \log m$, penalizes A with large m
- e.g. $g_R(D, A) = 1 \quad g_R(D, B) = 1$

Gini Index

- $\text{Gini}(D)$

$$
\begin{aligned}
\mathrm{Gini}(D) :&= \sum_{k=1}^{K} \frac{|C_k|}{|D|}(1 - \frac{|C_k|}{|D|}) \\
&= 1 - \sum_{k=1}^{K} (\frac{|C_k|}{|D|})^2 \\
&\leq 1 - \frac{1}{k}
\end{aligned}
$$

- approximates $\sum_{k=1}^{K} P(y=k)\big(1-P(y=k)\big)$

- $\mathrm{Gini}(D, A)$

$$
\mathrm{Gini}(D, A) := \sum_{i=1}^{m} \frac{|D_i|}{|D|} \mathrm{Gini}(D_i)
$$

- $\arg\min_A \mathrm{Gini}(D, A)$
- don't need to normalize

L2 loss

- For regression we just use L2 loss to measure purity

$$
\bar{y}_{D_i} = \frac{1}{|D_i|} \sum_{j \in D_i} y_j
$$

- $L(D, A)$

$$
L(D, A) = \sum_{i=1}^{m} \big[\sum_{j \in D_i} (y_i - \bar{y}_{D_i})^2 \big]
$$

- $\arg\min_A L(D, A)$

Build a Tree

Global Optimum

- d features, every feature has m values
- $f(d) = d(f(d-1))^m$
- This is a NP-hard problem!

Greedy Algorithm

- Given D, feature set F, choose a feature A according to some purity scores, partition D into $D_1, \ldots, D_m$
- Recursively build a tree for each D_i, with $F = F - \{A\}$
- When $F = \emptyset$ or D_i only contains a single class(pure), then stop and create a leaf node
  - with its majority label as prediction
  - with mean $\bar{y}_{D_i}$ as prediction for regression
- Regularization (early stop)
  - reach a maximize height
  - information gain $\leq \epsilon$
  - validation accuracy no longer increases
- Tree models: **low bias, high variance**