FADML Scribe

Neeraj Ahire (24BM6JP36)

6th March 2025

1 Decision Tree Learning (RECAP)

1.1 Measures of Impurity:

Consider a Binary classification problem, with two discrete classes, positive and negative. Following are the measures of impurity.

1.1.1 Entropy

The entropy of a dataset S is defined as:

$$E(S) = -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

where:

- S is the dataset.
- $p_{\oplus}$ is the probability of an instance belonging to the positive class.
- $p_{\ominus}$ is the probability of an instance belonging to the negative class.

1.1.2 Gini Index

The Gini Index of a dataset S is defined as:

$$G(S) = 1 - p_{\oplus}^2 - p_{\ominus}^2$$

where:

- S is the dataset.
- $p_{\oplus}$ is the probability of an instance belonging to the positive class.
- $p_{\ominus}$ is the probability of an instance belonging to the negative class.

1.2 Information Gain and Gini impurity

The Information Gain and Gini Impurity for an attribute A is given by:

$$Information\ Gain(S, A) = E(S) - \sum_{v \in V} \frac{|S_v|}{|S|} E(S_v)$$

$$Gini\ Impurity(S, A) = G(S) - \sum_{v \in V} \frac{|S_v|}{|S|} G(S_v)$$

where:

- $IG_E(S, A)$ is the information gain using entropy.
- $IG_G(S, A)$ is the information gain using Gini index.

- $E(S)$ is the entropy of the dataset before the split.
- $G(S)$ is the Gini index of the dataset before the split.
- S_v is the subset where attribute A has value v.
- $\frac{|S_v|}{|S|}$ represents the proportion of S_v in S.
- $E(S_v)$ is the entropy of subset S_v.
- $G(S_v)$ is the Gini index of subset S_v.

In decision trees, **Information Gain (IG)** uses **Entropy** to decide the best attribute for splitting a node. **Information Gain** measures how much uncertainty(entropy) is reduced after a split, selecting the attribute that provides the highest reduction in disorder. **Gini Index** evaluates impurity by selecting the attribute that results in the purest child nodes. While **Entropy is more precise**, **Gini is computationally faster**, and both aim to create the most informative splits for better classification.

1.3 Classification Boundary in Decision Trees (Visualization)

We use measures of impurity such Entropy or Gini index and then the Information gain or Gini impurity criteria build a Decision Tree.
Following is a visual depiction of classification using decision trees.

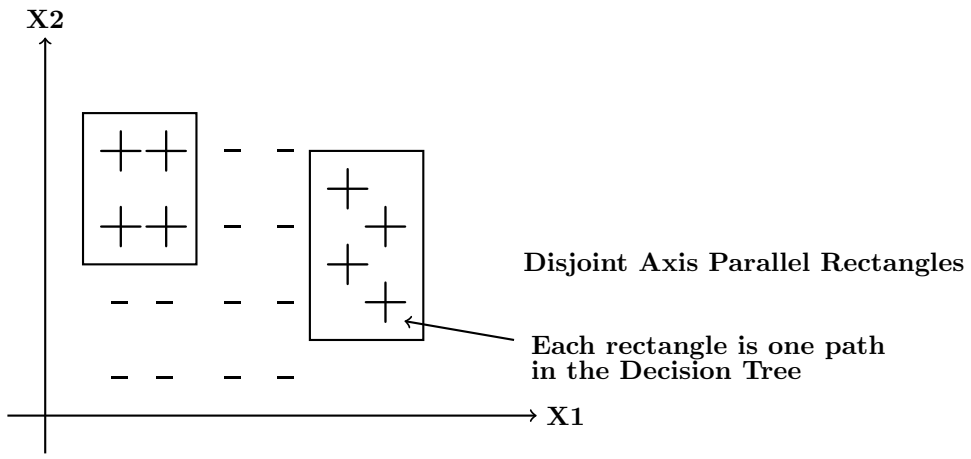

Figure 1: Disjoint Axis-Parallel Rectangles in Decision Tree Splitting

In decision tree classification, disjoint axis-parallel rectangles represent regions in the feature space where the decision tree partitions data based on learned rules. Each rectangle corresponds to a leaf node in the decision tree, containing instances that share the same predicted class. These rectangles are axis-aligned, meaning splits occur along feature axes (X1, X2, etc.), creating non-overlapping decision regions. This structure allows decision trees to model non-linear decision boundaries effectively. The diagram illustrates how two disjoint rectangles capture positive (+) class regions, while the remaining space is assigned to the negative (-) class, demonstrating how decision trees recursively divide the feature space.

2 Probabilistic (Bayesian) Approach to a Learning Problem

For a typical learning problem, we approximate the unknown function **g** using **f**. For example, consider below for the two class classification problem with classes +1 and -1 and attributes **X**.

$$g \approx \mathrm{f}: X \to Y \begin{cases} +1 \\ -1 \end{cases}$$

The **probabilistic (Bayesian) approach** to solving a **learning problem** relies on estimating **posterior probabilities** of different classes given the input features using **Bayes' theorem**. This is contrary to Decision Tree learning where we have a fixed output instead of probabilities. Classification is performed based on the **likelihood ratio**:

$$\frac{P(y=1 \mid X_1, \ldots, X_n)}{P(y=0 \mid X_1, \ldots, X_n)} > 1 \; or < 1$$

meaning the instance is classified into the class with the **higher posterior probability**. This approach incorporates **uncertainty** and is particularly useful in cases of **limited data** or **noisy environments**.

2.1 Baye's Rule

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}$$

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

Generalization of Baye's Rule:

$$P(Y \mid X_1, \ldots, X_n) = \frac{P(Y)P(X_1, \ldots, X_n \mid Y)}{P(X_1, \ldots, X_n)}$$

$$P(Y \mid X_1, \ldots, X_n) = \frac{P(Y)P(X_1, \ldots, X_n \mid Y)}{P(Y)P(X_1, \ldots, X_n \mid Y) + P(\bar{Y})P(X_1, \ldots, X_n \mid \bar{Y})}$$

Note the following rules:

$$P(A = 1 \mid X) = 1 - P(A = 0 \mid X)$$

$$P(X \mid A = 1) \neq 1 - P(X \mid A = 0)$$

2.2 Joint Probability Distribution Table

A **joint distribution table (JPDT)** represents the **probabilities of different combinations** of random variables occurring together. It helps in **computing conditional probabilities** using **Bayes' theorem** and is essential for **learning probabilistic models**. For any kind of outcome we could just do lookup in the table in order to predict\estimate. Refer the example table below.

Gender (X1)	Hours Worked (X2)	Poor/Rich (Y)	Prob
M	> 40	R	0.15
M	> 40	P	0.2
M	< 40	R	0.1
M	< 40	P	0.3
F	< 40	R	0.1
F	> 40	P	0.05
F	< 40	R	0.05
F	< 40	P	0.05

Consider now, the example below to get the probability of 'Rich' if 'Gender' is female and 'Hours Worked' is less than 40-

$$P(R \mid F, < 40) = \frac{P(R, F, < 40)}{P(F, < 40)}$$

$$= \frac{0.05}{0.1} = \frac{1}{2}$$

2.3 Issue of Data Sparsity

If the number of attributes is n, there will be $2^n + 1$ entries in the joint distribution table that we need for estimation/prediction. This number grows exponentially. For example, if we have 100 attributes, the number of entries in the table is approximately 2^{100}, and it is unlikely that we will have all the data, leading to the problem of **data sparsity**.

We use the following approaches to tackle this problem:

1. Smart Estimation: refers to efficiently estimating probabilities in a joint distribution table, especially when data is sparse.

2. Smart Representation: focuses on storing and structuring the joint probability table efficiently, reducing memory and computational costs.

2.4 Estimation

2.4.1 Maximum Likelihood Estimation (MLE)

Maximum Likelihood Estimation (MLE) is a method for estimating the parameter θ of a probability distribution by maximizing the **likelihood function** $P(\text{Data} \mid \theta)$. In the context of **Bayesian learning**, MLE does not incorporate prior knowledge; it solely relies on observed data.

$$\theta_{\text{MLE}} = \arg\max_{\theta} P(\text{Data} \mid \theta)$$

Where:

- θ_{MLE} : The **Maximum Likelihood Estimate**, which is the parameter value θ that maximizes the probability of the observed data.
- $P(\text{Data} \mid \theta)$: The **likelihood function**, which quantifies how well the parameter θ explains the observed data.
- $\arg\max_\theta$: Represents the value of θ that gives the highest likelihood.

Derivation of MLE for a simple coin toss problem:

$$\frac{\partial}{\partial\theta}\left[\theta^{\alpha_H}(1-\theta)^{\alpha_T}\right] = 0$$

$$\Rightarrow \frac{\partial}{\partial\theta}\left[\ln\left(\theta^{\alpha_H}(1-\theta)^{\alpha_T}\right)\right] = 0$$

$$\Rightarrow \frac{\partial}{\partial\theta}\left[\alpha_H \ln\theta + \alpha_T \ln(1-\theta)\right] = 0$$

$$\Rightarrow \frac{\alpha_H}{\theta} - \frac{\alpha_T}{1-\theta} = 0$$

$$\Rightarrow (1-\theta)\alpha_H = \theta\alpha_T$$

$$\Rightarrow \hat{\theta}_{\text{MLE}} = \frac{\alpha_H}{\alpha_H + \alpha_T}$$

$$\Rightarrow \hat{\theta}_{\text{MLE}} = \frac{\alpha_H}{\alpha_H + \alpha_T}$$

where α_H represents the number of observed heads, α_T represents the number of observed tails, and θ represents the probability of getting heads in a single coin toss.

2.4.2 Maximum A Posteriori (MAP)

Maximum A Posteriori (MAP) estimation is a Bayesian approach to parameter estimation that finds the **most probable value** of a parameter θ given both the observed data and prior knowledge. Unlike **Maximum Likelihood Estimation (MLE)**, which maximizes only the likelihood function $P(\text{Data} \mid \theta)$, MAP also incorporates a **prior distribution** $P(\theta)$.

MAP estimation is particularly useful when **data is limited** or **noisy**. In contrast to MLE, which can be unstable with small datasets, MAP provides a **balance between data-driven learning and prior knowledge**.

$$\theta_{\text{MAP}} = \arg\max_{\theta} P(\theta \mid \text{Data})$$

Using Bayes' theorem, we expand it as:

$$\theta_{\text{MAP}} = \arg\max_{\theta} \frac{P(\text{Data} \mid \theta)P(\theta)}{P(\text{Data})}$$

Since $P(\text{Data})$ **is constant for all** θ, we simplify to:

$$\theta_{\text{MAP}} = \arg\max_{\theta} P(\text{Data} \mid \theta)P(\theta)$$

Where:

- θ_{MAP} is the **Maximum A Posteriori (MAP) estimate** of the parameter θ.
- $P(\theta \mid \text{Data})$ is the **posterior distribution**, representing the probability of θ given the observed data.
- $P(\text{Data} \mid \theta)$ is the **likelihood function**, measuring how well θ explains the data.
- $P(\theta)$ is the **prior probability**, representing our belief about θ before observing the data.
- $P(\text{Data})$ is the **marginal likelihood** (or evidence), a normalizing constant that ensures the posterior is a valid probability distribution.

The MAP estimate for the probability of heads in a simple coin toss problem θ in a Bernoulli/Binomial setting with a Beta prior is given by:

$$\theta_{\text{MAP}} = \frac{\alpha_H + H}{\alpha_H + \alpha_T + H + T}$$

Derivation:

$$\frac{\partial}{\partial\theta}\left[\theta^{\alpha_H}(1-\theta)^{\alpha_T}\left(\frac{\theta^{\beta_H}(1-\theta)^{\beta_T}}{B(\beta_H,\beta_T)}\right)\right] = 0$$

$$\Rightarrow \frac{\partial}{\partial\theta}\left[\theta^{\alpha_H+\beta_H}(1-\theta)^{\alpha_T+\beta_T}\right] = 0$$

Expanding further leads to θ_{MAP} above.

Where:

- θ_{MAP} : The **Maximum A Posteriori estimate** of θ, incorporating prior information.
- H : The number of observed heads.
- T : The number of observed tails.
- α_H : The prior count (pseudo-count) of heads.
- α_T : The prior count (pseudo-count) of tails.