CS217: Artificial Intelligence and Machine Learning (associated lab: CS240)

Nihar Ranjan Sahoo
PhD scholar under Prof. Pushpak Bhattacharyya
CSE Dept.,
IIT Bombay

Week11 of 24mar25, Decision Trees, Intro Speech Recognition

Decision Tree

Match	Pitch Type	Host	Batting First	Winner
M1	Spin-friendly	India	India	India
M2	Pace-friendly	Australia	Australia	Australia
M3	Balanced	India	Australia	India
M4	Spin-friendly	Australia	India	Australia
M5	Pace-friendly	India	Australia	Australia
M6	Spin-friendly	India	Australia	India
M7	Balanced	Australia	India	India
M8	Pace-friendly	Australia	India	Australia
M9	Spin-friendly	India	India	India
M10	Balanced	Australia	Australia	Australia

1. Make decision to predict if India can beat Australia in the upcoming match?
    a. Use Information Gain/Gain Ratio/Gini Index to build decision tree.

Decision Tree

Entropy == Randomness

$$X \sim \mathrm{Bern}(P)$$

$$P(x=1) = P$$

$$P(x=0) = 1 - P$$

$$H(x) = \sum_{z=1}^{k} -P(x=i)\log P(x=i)$$

$$= -\int_{-\infty}^{\infty} P(x)\log P(x)dx$$

$$= E_{x\sim P(x)}[-\log P(x)]$$

Find *P* that maximizes the entropy for a Bern(P) => MLE

$$H(\mathrm{Bern}\, n(P)) = -P\log P - (1-P)\log(1-P)$$

$$\frac{\partial H}{\partial P} = \frac{-P}{P} - \log P + \log(1-P)$$

$$H = 0$$

$$\log\left(\frac{1-P)}{P}\right) = 0$$

$$\Rightarrow 1 - P = P$$

$$\Rightarrow P = 1/2$$

$$H(\mathrm{Bern}(1/2)) = -1/2\log 1/2 - (1-1/2)\log(1-1/2)$$

$$H(\mathrm{Bern}(1/2)) = (1/2)\log 2 + (1/2)\log 2 = \log 2$$

$$\lim_{P\to 0^+} -P\log P - (1-P)\log(1-P)$$

$$\lim_{P\to 0^+} \frac{-\log P}{1/P} = \frac{-\log P}{-1/P} = \lim_{P\to 0^+} P = 0$$

$$\text{Similarily, } \lim_{P\to 1^-} H(P) = 0$$

What needs to be decided on?

- Split feature
    - based on Purity on feature
- Split point
- When to stop splitting

Purity:

- how homogeneous a node is in terms of class labels
- goal of splitting is to **create child nodes that are purer** than the parent node
- meaning they contain more instances of a single class

Different Purity measures?

1. Gini Impurity

$$Gini = 1 - \sum_{i=1}^{c} p_i^2$$

- Measures the probability of incorrectly classifying a randomly chosen element.
- Lower values indicate purer nodes.
- Used in **CART (Classification and Regression Trees)**.

2. Entropy (Information Gain)

$$Entropy = -\sum_{i=1}^{c} p_i \log_2 p_i$$

- Measures the uncertainty in a node.
- Used in **ID3, C4.5, and C5.0** algorithms.
- A split is chosen to maximize **Information Gain**:

$$IG = Entropy(parent) - \sum \frac{|child|}{|parent|} \times Entropy(child)$$

Different Purity measures?

Variance Reduction (for Regression Trees)

$$\text{Variance} = \frac{1}{N}\sum(y_i - \bar{y})^2$$

- Used for regression tasks.
- The split is chosen to minimize variance within child nodes.

Splitting Strategy- At each step, the algorithm:

- Evaluates all possible splits.
- Computes the purity measure for each split.
- Selects the split that results in the highest improvement in purity.

Information Gain to construct Decision Tree

Match	Pitch Type	Host	Batting First	Winner
M1	Spin-friendly	India	India	India
M2	Pace-friendly	Australia	Australia	Australia
M3	Balanced	India	Australia	India
M4	Spin-friendly	Australia	India	Australia
M5	Pace-friendly	India	Australia	Australia
M6	Spin-friendly	India	Australia	India
M7	Balanced	Australia	India	India
M8	Pace-friendly	Australia	India	Australia
M9	Spin-friendly	India	India	India
M10	Balanced	Australia	Australia	Australia

Information Gain to construct Decision Tree

$$Entropy(S) = -p_1 \log_2 p_1 - p_2 \log_2 p_2$$

where p_1 and p_2 are the probabilities of **India winning** and **India not winning (Australia winning)**.

From the table:

- **Total matches** = 10
- **India wins** = 5
- **Australia wins** = 5

$$p(India) = \frac{5}{10} = 0.5, \quad p(Australia) = \frac{5}{10} = 0.5$$

$$Entropy(S) = -\left(0.5 \log_2 0.5 + 0.5 \log_2 0.5\right)$$

$$= -(0.5 \times -1 + 0.5 \times -1)$$

$$= -(-0.5 - 0.5) = 1.0$$

Entropy for Spin

$$p(India) = \frac{3}{4}, \quad p(Australia) = \frac{1}{4}$$

$$Entropy(Spin) = -\left(\frac{3}{4} \log_2 \frac{3}{4} + \frac{1}{4} \log_2 \frac{1}{4}\right)$$

$$= -(0.75 \times -0.415 + 0.25 \times -2)$$

$$= -(-0.311 - 0.5) = 0.811$$

Information Gain to construct Decision Tree

Entropy for Pace

$$p(India) = 0, \quad p(Australia) = 1$$

$$Entropy(Pace) = -(0\log_2 0 + 1\log_2 1) = 0$$

$$p(India) = \frac{2}{3}, \quad p(Australia) = \frac{1}{3}$$

Entropy for Balanced

$$Entropy(Balanced) = -\left(\frac{2}{3}\log_2\frac{2}{3} + \frac{1}{3}\log_2\frac{1}{3}\right)$$

$$= -(0.667 \times -0.585 + 0.333 \times -1.585)$$

$$= -(-0.390 - 0.528) = 0.918$$

Weighted Entropy of Pitch

$$Entropy(Pitch) = \frac{4}{10} \times 0.811 + \frac{3}{10} \times 0 + \frac{3}{10} \times 0.918$$

$$= 0.3244 + 0 + 0.2754 = 0.5998$$

Information Gain

$$IG = Entropy(S) - Entropy(Pitch)$$

$$IG = 1.0 - 0.5998$$

$$IG = 0.4002$$

IG(Host)=1.0−0.971=0.029. | *IG(Batting)=1.0−0.971=0.029*

Stopping Criteria in Decision Tree

1. Pure Node

1. No significant IG

1. Minimum Samples in a node

1. Maximum tree depth

1. No features to split

Definition of a linear model

A linear model is considered **linear** because the model's predictions are a **linear function** of the parameters w.

Mathematically, a typical linear model takes the form:

$y=w^Tx+b$

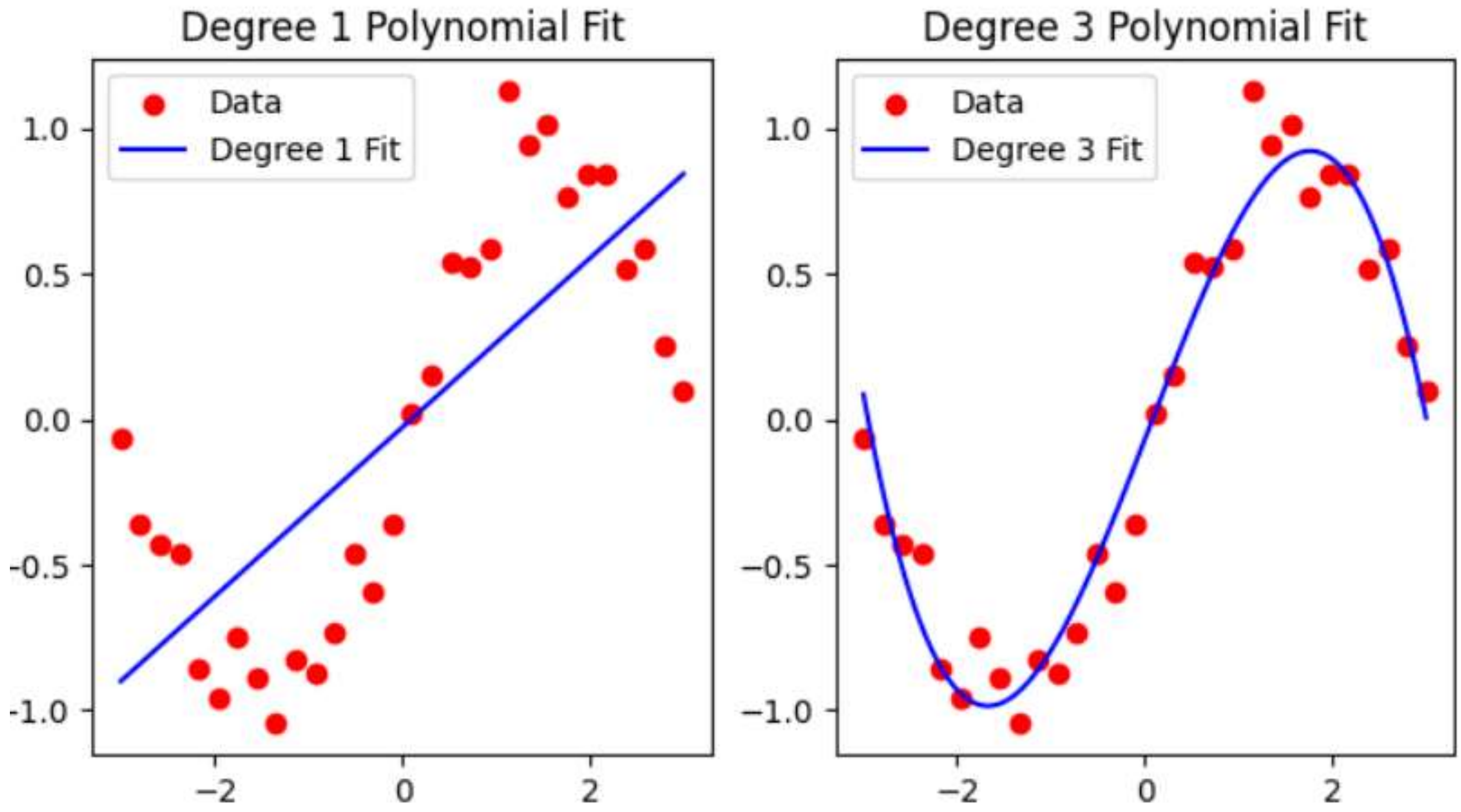
where:

- x is the input feature vector,
- w is the weight vector (parameters),
- b is the bias term,
- y is the predicted output.

Bias-Variance Tradeoff: Overfitting and Underfitting

Overfitting: The model learns not only the underlying pattern but also the noise in the training data. It performs well on training data but poorly on unseen data.

Underfitting: The model is too simple to capture the underlying pattern in the data, leading to poor performance on both training and test data.



Bias-Variance Tradeoff: Overfitting and Underfitting

Overfitting: The model learns not only the underlying pattern but also the noise in the training data. It performs well on training data but poorly on unseen data.

Underfitting: The model is too simple to capture the underlying pattern in the data, leading to poor performance on both training and test data.