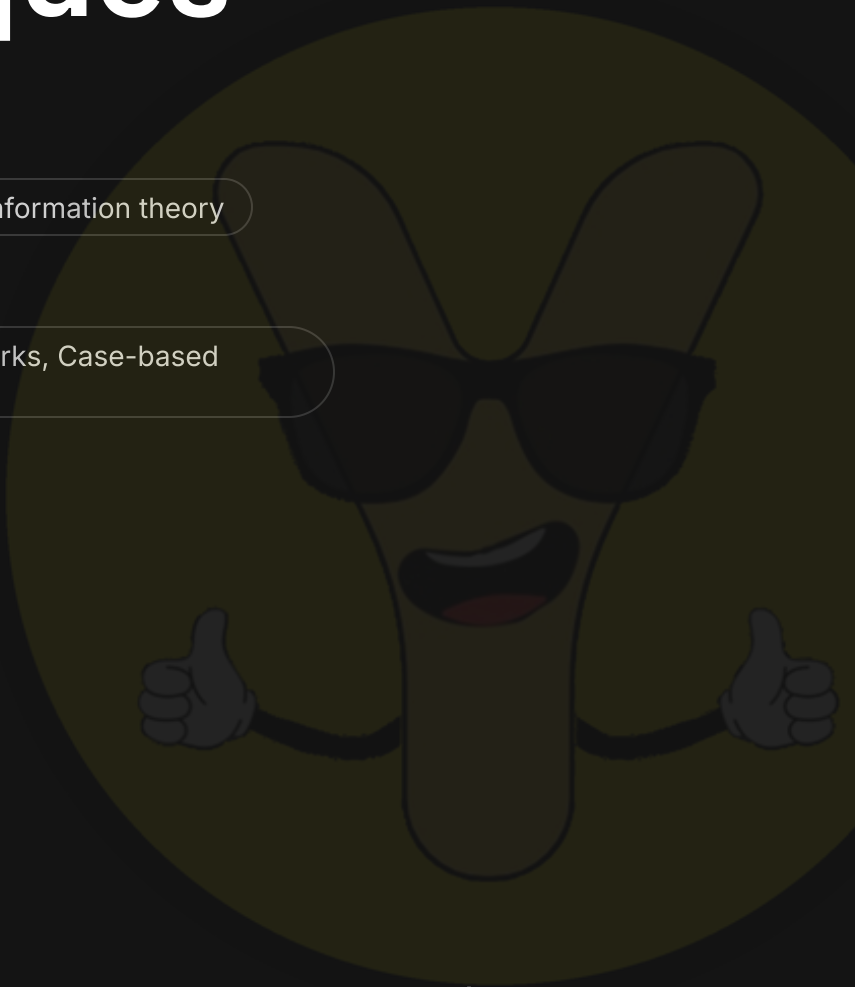
AKTU · 3RD YEAR · SEMESTER 6 · BCDS062

Machine Learning Techniques

Unit 3 — Decision Tree and Instance-Based Learning

- Decision tree learning algorithm
- Inductive bias
- Inductive inference with decision trees
- Entropy and information theory
- Information gain
- ID-3 Algorithm
- Issues in Decision tree learning
- Instance-based learning: k-Nearest Neighbour Learning, Locally Weighted Regression, Radial basis function networks, Case-based learning



DECODED BY

Chirag Jain

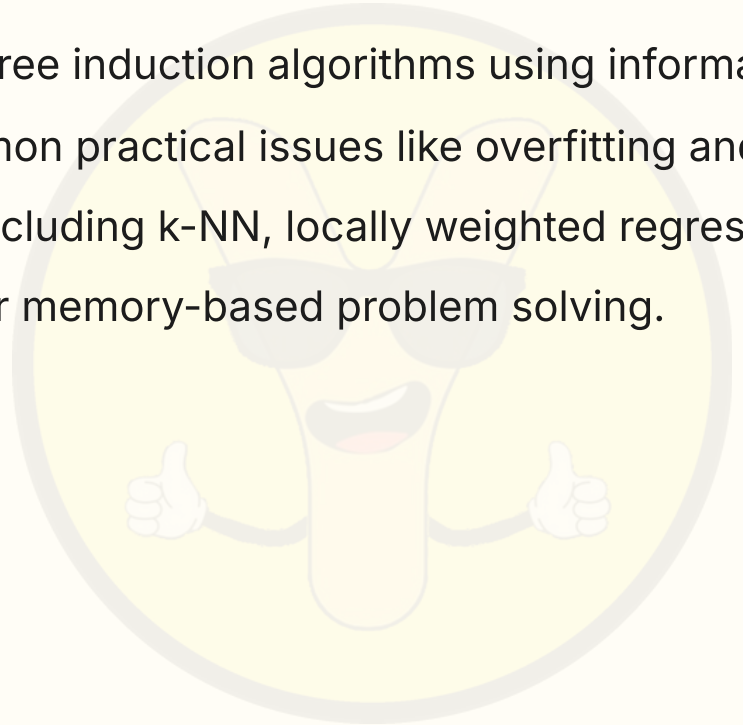
Founder, Ycotes · Gold Medalist in Artificial Intelligence, IIT Madras · taught 10,000+ students

theycotes.com

Unit 3: Decision Tree and Instance-Based Learning

Introduction to non-parametric supervised learning techniques for classification and regression.

- Covers greedy top-down decision tree induction algorithms using information theory.
- Addresses inductive bias and common practical issues like overfitting and pruning.
- Explores lazy learning paradigms including k-NN, locally weighted regression, and RBF networks.
- Examines case-based reasoning for memory-based problem solving.



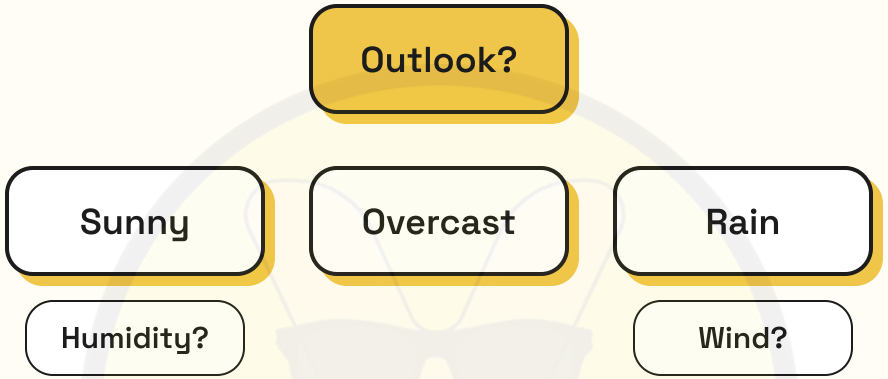
Decision Tree Learning Algorithm

DECISION TREE

A supervised learning algorithm that approximates discrete-valued target functions by constructing a flowchart-like tree structure.

- Internal nodes represent tests on specific input attributes.
- Branches denote the possible outcomes of those attribute tests.
- Leaf nodes hold the final predicted class labels or continuous values.
- Operates via a greedy top-down recursive divide-and-conquer strategy.

Decision Tree Architecture



A sample decision tree classifying whether to play tennis based on weather conditions.

Inductive Bias in Decision Trees

INDUCTIVE BIAS

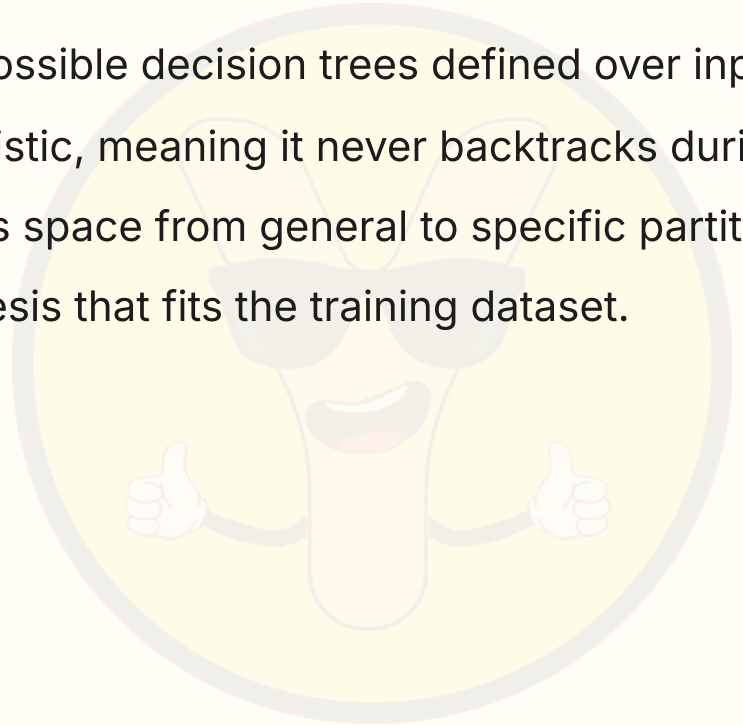
The set of assumptions that a learning algorithm uses to predict outputs for unseen data.

- ID3 and C4.5 prefer shorter trees over longer trees.
- They prioritize trees that place attributes with high information gain closer to the root.
- Employs a preference bias rather than a strict restriction bias.
- Rooted in Occam's Razor: simpler hypotheses are preferred over complex ones.

Inductive Inference with Decision Trees

How decision trees search the hypothesis space to generalize from training instances.

- Hypothesis space is the set of all possible decision trees defined over input attributes.
- Search strategy is greedy and heuristic, meaning it never backtracks during construction.
- Traverses an incomplete hypothesis space from general to specific partitions.
- Outputs a single consistent hypothesis that fits the training dataset.



Entropy and Information Theory

ENTROPY

A measure of impurity, disorder, or uncertainty in a given collection of training examples.

- Defined mathematically as $H(S) = -\sum_{i=1}^{c} p_i \log_2 p_i$.
- Here, S is the dataset and p_i is the proportion of class i in S.
- Entropy is 0 when all examples belong to the same class (pure node).
- Entropy is 1 (for binary classification) when classes are evenly split (maximum impurity).

Information Gain

INFORMATION GAIN

The expected reduction in entropy achieved by partitioning the dataset according to a specific attribute.

- Calculated as $Gain(S, A) = H(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} H(S_v)$.
- S_v represents the subset of S where attribute A has value v.
- Quantifies the effectiveness of an attribute in classifying the training data.
- Attributes with higher information gain are selected as decision nodes first.

Calculating Entropy and Information Gain

Dataset has 14 examples: 9 Yes, 5 No. Split by attribute Wind (Weak: 6 Yes, 2 No; Strong: 3 Yes, 3 No). Find Gain(S, Wind).

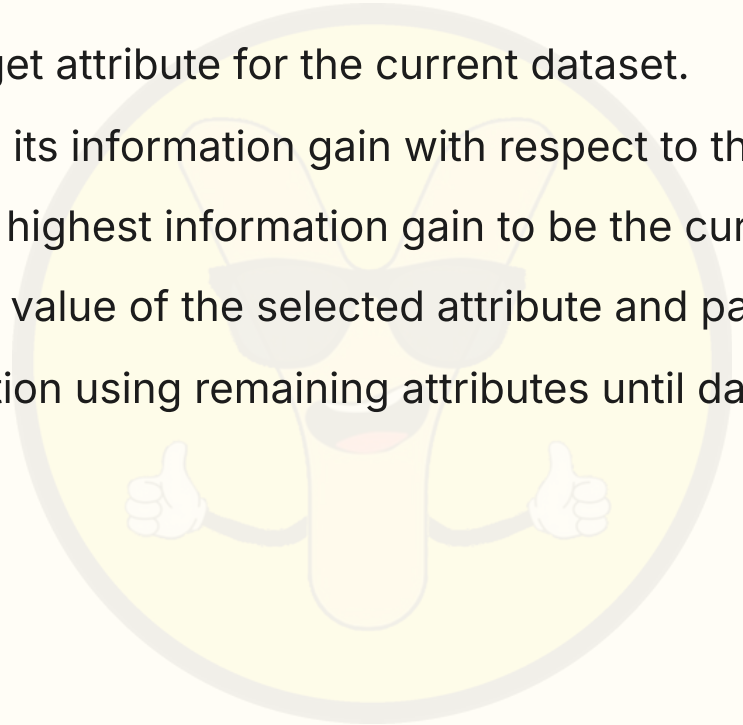
1. Calculate total entropy: $H(S) = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14) = 0.940$.
2. Calculate entropy for Weak: $H(S_{weak}) = -(6/8)\log_2(6/8) - (2/8)\log_2(2/8) = 0.811$.
3. Calculate entropy for Strong: $H(S_{strong}) = -(3/6)\log_2(3/6) - (3/6)\log_2(3/6) = 1.000$.
4. Compute Gain: $Gain(S, Wind) = 0.940 - [(8/14)(0.811) + (6/14)(1.000)]$.

Gain(S, Wind) = 0.940 - 0.892 = 0.048 bits.

ID3 Algorithm

Iterative Dichotomiser 3 is the foundational algorithm for building decision trees.

- Step 1: Calculate entropy of the target attribute for the current dataset.
- Step 2: For each attribute, calculate its information gain with respect to the dataset.
- Step 3: Select the attribute with the highest information gain to be the current node.
- Step 4: Create child nodes for each value of the selected attribute and partition data.
- Step 5: Recurse on each child partition using remaining attributes until data is pure.



Issues in Decision Tree Learning

Practical challenges encountered when deploying decision trees in real-world applications.

- Overfitting
  - Trees grow too complex, fitting noise in training data while performing poorly on test data.
  - Addressed via pre-pruning (stopping early) and post-pruning (reducing subtrees).
- Continuous Attributes
  - ID3 handles only discrete values; continuous attributes require threshold discretization.
- Missing Attribute Values
  - Handled by assigning the most common value overall or distributing fractional counts.
- Attribute Bias
  - Information gain favors attributes with many distinct values (e.g., ID numbers).

Instance-Based Learning

LAZY LEARNING

A family of learning algorithms that defer generalization until a classification query is received.

- Does not construct an explicit global abstraction or model during training.
- Stores all training instances in memory and relies on local distance metrics.
- Also known as lazy learning, contrasted with eager learners like decision trees.
- High training time is zero, but testing and query time can be very expensive.

k-Nearest Neighbour (k-NN) Learning

K-NN ALGORITHM

A non-parametric instance-based method that classifies unlabelled points based on the majority vote of their k-nearest neighbors.

- Uses Euclidean distance to measure proximity: $d(x, y) = \sqrt{\sum (x_i - y_i)^2}$.
- For classification, takes a majority vote among the k closest training examples.
- For regression, computes the average of the target values of the k nearest neighbors.
- Sensitive to the choice of k and feature scaling (normalization is mandatory).

Locally Weighted Regression (LWR)

LOCALLY WEIGHTED REGRESSION

A generalisation of linear regression that constructs an explicit local approximation for each query point.

- Weights training examples based on their distance from the query point using a kernel function.
- Points closer to the query point receive higher weights, while distant points receive lower weights.
- Solves a weighted least squares problem specifically tailored for the query point.
- Computationally expensive because a separate regression model is solved for every query.

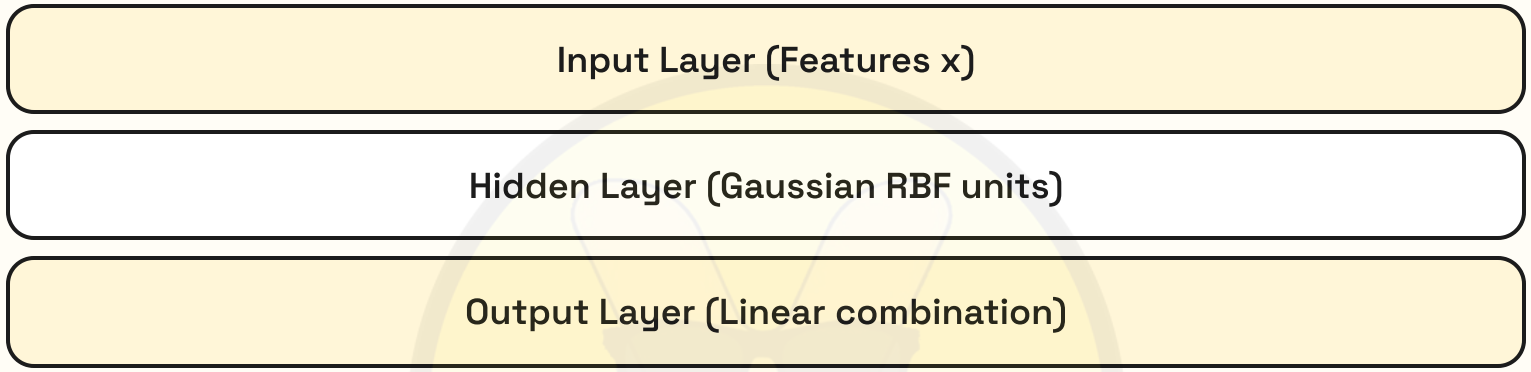
Radial Basis Function (RBF) Networks

RBF NETWORK

An artificial neural network that uses radial basis functions as activation functions for hidden units.

- Consists of three layers: Input layer, Hidden layer with non-linear RBF units, and Linear output layer.
- Hidden units compute distance between input vector and a center vector, typically using Gaussian kernels.
- Gaussian activation: $h_j(x) = \exp(-\frac{||x-\mu_j||^2}{2\sigma_j^2})$.
- Combines aspects of instance-based learning and neural networks for fast training.

RBF Network Architecture



Three-layer architecture of a Radial Basis Function Network.

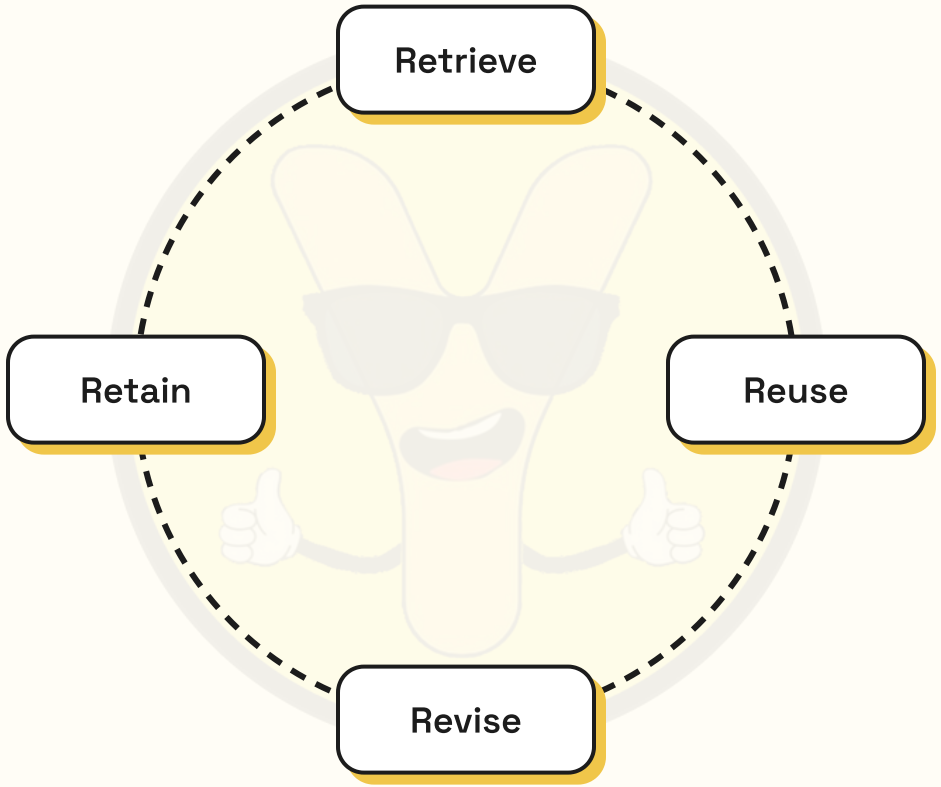
Case-Based Reasoning (CBR)

CASE-BASED REASONING

An incremental learning paradigm that solves new problems by retrieving and adapting previously successful past cases.

- Operates on the R4 cycle: Retrieve, Reuse, Revise, and Retain.
- Unlike pure k-NN, CBR cases can be complex symbolic descriptions, plans, or structured text.
- Requires explicit domain knowledge to adapt solutions and handle edge cases.
- Widely used in expert systems, customer support helpdesks, and medical diagnosis.

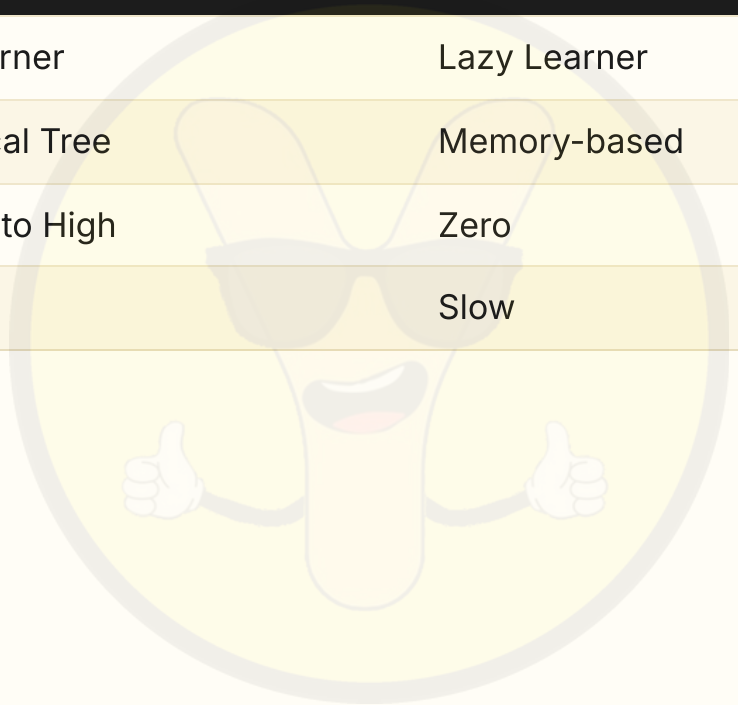
The CBR Cycle



The four-step process of Case-Based Reasoning.

Comparison of Learning Paradigms

Feature	Decision Trees	k-NN Learning	RBF Networks
Learning Type	Eager Learner	Lazy Learner	Eager Learner
Model Structure	Hierarchical Tree	Memory-based	Neural Network
Training Time	Moderate to High	Zero	Moderate
Testing Time	Fast	Slow	Fast



Exam corner

- **Q1** Explain the ID3 decision tree algorithm with a suitable numerical example calculating entropy and information gain.
- **Q2** What is inductive bias? Compare the inductive bias of decision tree learning with candidate elimination algorithm.
- **Q3** Discuss the problem of overfitting in decision trees and explain pre-pruning and post-pruning strategies.
- **Q4** Explain k-Nearest Neighbour algorithm. How does choice of distance metric and value of k affect its performance?
- **Q5** Derive or explain Locally Weighted Regression and contrast it with standard linear regression.
- **Q6** Explain the architecture and working principle of Radial Basis Function (RBF) networks.
- **Q7** What is Case-Based Reasoning? Explain the R4 cycle with a neat diagram.

Unit 3 done. You are exam-ready.

Revise the Exam corner once more, then open Unit 4 on theycotes.com. Every AKTU subject, unit by unit, free. Notes by Chirag Jain, Ycotes.

