Kalinga University

Department of Computer Science & Information Technology

Course- BCAAIML
Subject- Machine Learning Basics
Subject Code – BCAAIML404 **Sem.- IV**

Unit II

Parameter Estimation

Objective: Understand how to estimate unknown parameters of a statistical model from data.

1.1 Definition and Importance

- **Parameter Estimation:** A process used to determine the parameters of a population distribution or model based on sample data.
  - ***Example:*** Estimating the mean and standard deviation of a normal distribution from a dataset.
- **Importance:**
  - Forms the foundation for statistical inference.
  - Crucial in fields like machine learning, econometrics, and biostatistics.

1.2 Types of Parameter Estimation

1. **Point Estimation**:
   - **Definition:** Provides a single value as the estimate of the unknown parameter.
   - **Properties of Good Estimators:**
     - **Unbiasedness:** The expected value of the estimator equals the true parameter value.
     - **Consistency:** The estimator converges to the true parameter as the sample size increases.
     - **Efficiency:** The estimator has the smallest variance among all unbiased estimators.
   - **Methods of Point Estimation:**
     - **Maximum Likelihood Estimation (MLE):**
       - **Concept:** Chooses parameters that maximize the likelihood function L(θ)L(\theta)L(θ) given the data XXX.
       - **Mathematical Formulation:**

$$\hat{\theta}_{MLE} = \arg\max_{\theta} L(\theta|X) = \arg\max_{\theta} \prod_{i=1}^{n} f(x_i|\theta)$$

-
  - **Example:** For a normal distribution N(μ,σ2)N(\mu, \sigma^2)N(μ,σ2), the MLE for the mean is the sample mean.

- **Method of Moments (MoM):**
  - **Concept:** Equates sample moments (mean, variance) to theoretical moments.
    - **Formulation:**

$$E[X^r] = \frac{1}{n}\sum_{i=1}^{n} x_i^r$$

-
  -
    -
    - **Example:** Estimating the mean of a population by equating it to the sample mean.

- **Interval Estimation**:
  - **Definition:** Provides a range of values within which the parameter is likely to lie.
  - **Confidence Interval (CI):**
    - **Interpretation:** A 95% confidence interval means that if the experiment is repeated 100 times, approximately 95 of the intervals will contain the true parameter.
    - **Calculation:** For a normal distribution with known variance:

$$\mu \in \left(\bar{x} - z_{\frac{\alpha}{2}}\frac{\sigma}{\sqrt{n}}, \bar{x} + z_{\frac{\alpha}{2}}\frac{\sigma}{\sqrt{n}}\right)$$

1.
   -
     - **pplications:** Hypothesis testing and decision-making.

1.3 Key Techniques

1. **Bayesian Estimation:**
    - **Concept:** Incorporates prior knowledge through a ***prior distribution*** and updates it with data using Bayes' Theorem.
    - **Posterior Distribution:**

$$P(\theta|X) = \frac{P(X|\theta)P(\theta)}{P(X)}$$

Applications: Medical diagnostics, machine learning (e.g., Naive Bayes classifiers).

- **Least Squares Estimation:**

- **Concept:** Minimizes the sum of the squared differences between observed and predicted values.
- **Common Use:** Linear regression.
    - **Formula:**

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

1.
    - **Example:** Predicting housing prices based on various features.

1.4 Real-world Examples

- **Engineering:** Estimating system parameters for control models.
- **Finance:** Calculating risk parameters in portfolio management.
- **Biostatistics:** Estimating disease prevalence rates from sample data.

Sufficient Statistics in Machine Learning

Objective: Understand how sufficient statistics simplify and enhance learning models by summarizing data effectively.

2.1 Definition and Role in Machine Learning

- **Sufficient Statistic in ML:** A function of the data that retains all information needed to estimate a model parameter efficiently.
    - **Relevance:** By reducing data without losing critical information, sufficient statistics help models learn faster and avoid redundant computations.

2.2 Factorization Theorem in ML Context

- **Concept:** Enables reducing data while maintaining its information content, which is critical for tasks like parameter estimation in probabilistic models.
    - **Practical Example:** In Bayesian networks, sufficient statistics allow summarization of large datasets before updating posterior probabilities.

2.3 Examples in Machine Learning Models

1. **Gaussian Mixture Models (GMM):**
    - For a Gaussian distribution N(μ,σ2)N(\mu, \sigma^2)N(μ,σ2), the sample mean x¯\bar{x}x¯ and variance s2s^2s2 are sufficient for estimating parameters μ\muμ and σ2\sigma^2σ2.
    - **Impact:** Reduces the dataset to these values, making EM (Expectation-Maximization) algorithms more efficient.
2. **Exponential Family Distributions:**
    - Many ML models (like logistic regression) use exponential family distributions where sufficient statistics simplify parameter learning.

      Factorization:

$$f(x|\theta) = h(x)\exp\left(\eta(\theta)^T T(x) - A(\theta)\right)$$

1.
    - **Example:** In multinomial classification, class counts serve as sufficient statistics.

2.4 Applications in Modern ML

- **Feature Extraction:** Simplifies complex datasets into minimal yet informative representations.
- **Dimensionality Reduction:** Supports efficient training in high-dimensional models by focusing only on sufficient summaries.

📚 3. Decision Trees in Machine Learning

Objective: Explore the role of decision trees in classification and regression tasks.

3.1 Overview and Structure

- **Definition:** A supervised learning algorithm that splits data into subsets based on feature values, forming a tree structure.
    - **Nodes:** Represent tests on features.
    - **Leaves:** Represent class labels (classification) or continuous values (regression).
- **Significance in ML:**
    - **Interpretable Models:** Easy to visualize and understand.
    - **Non-linear Relationships:** Capture complex patterns without requiring linearity assumptions.

3.2 Key Algorithms

1. **ID3 (Iterative Dichotomiser 3):**
    - Builds the tree by selecting the feature with the highest information gain.
2. **C4.5 and CART (Classification and Regression Trees):**
    - **C4.5:** Extends ID3 with handling for continuous data and missing values.
    - **CART:** Supports both classification and regression by minimizing Gini Impurity or mean squared error.

3.3 Practical Considerations

- **Overfitting:**
    - Occurs when trees are too deep and fit noise.
    - **Solution:** Pruning (removing unnecessary branches).
- **Feature Importance:** Decision trees provide insights into which features contribute most to predictions.

3.4 Real-World Applications

- **Medical Diagnosis:** Classifying diseases based on symptoms.
- **Finance:** Credit scoring models.
- **Data Preprocessing:** Often used as the base model in ensemble techniques like Random Forests and Gradient Boosting.

Introduction to Neural Networks

1.1 Definition and Concept

A neural network is a computational model inspired by the human brain's interconnected neurons. It consists of layers of artificial neurons (also called nodes) that process input data to generate outputs.

1.2 Why Neural Networks?

- **Pattern Recognition:** Capable of recognizing complex, non-linear patterns.
- **Universal Approximation:** Can approximate any continuous function given sufficient neurons and data.
- **Adaptability:** Learn from data, improving performance over time.

◆ 2. Architecture of Neural Networks

2.1 Basic Structure

A neural network consists of three main types of layers:

1. **Input Layer:**
   - Receives the input features (e.g., pixel values in an image).
2. **Hidden Layers:**
   - Perform intermediate computations and extract features.
   - Each hidden layer contains multiple neurons connected to the previous and next layers.
3. **Output Layer:**
   - Produces the final prediction (e.g., class label or regression value).

2.2 Neuron Model (Perceptron)

- **Mathematical Representation:**
  Each neuron computes a weighted sum of its inputs, adds a bias term, and applies an activation function:

$$z = w_1x_1 + w_2x_2 + \cdots + w_nx_n + b = \sum_{i=1}^{n} w_ix_i + b$$

$$y = \phi(z)$$

- where:
  - xix_ixi: Input features
  - wiw_iwi: Weights
  - bbb: Bias
  - ϕ(z)\phi(z)ϕ(z): Activation function

Activation Functions

3.1 Purpose of Activation Functions

- Introduce non-linearity into the model, allowing it to learn complex patterns.
- Common activation functions include:

1. **Sigmoid Function:**

$$\phi(z) = \frac{1}{1 + e^{-z}}$$

- • **Range:** (0, 1)
- **Use:** Binary classification.

- **ReLU (Rectified Linear Unit):**

$$\phi(z) = \max(0, z)$$

- • **Range:** [0, ∞)
- **Use:** Most common in hidden layers; helps mitigate the vanishing gradient problem.

- **Tanh Function:**

$$\phi(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

- • **ange:** (-1, 1)
- **Use:** Similar to Sigmoid but centered around 0.

- **Softmax Function:**

$$\phi(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}$$

1.
   - **Use:** Multi-class classification in the output layer.

◆ 4. Forward Propagation

Concept:

Forward propagation computes the output by passing inputs through the network layer by layer.

1. **Input Layer:**
   Takes the raw input vector XXX.
2. **Hidden Layers:**
   Each neuron in a hidden layer computes:

$$z^{(l)} = W^{(l)} X^{(l-1)} + b^{(l)}$$

$$a^{(l)} = \phi(z^{(l)})$$

1. **Output Layer:**
   The final output depends on the activation function (e.g., Softmax for classification).

◆ 5. Training Neural Networks

5.1 Objective:

Minimize the difference between the predicted output and the actual target (loss function).

5.2 Loss Functions

- **Mean Squared Error (MSE):** For regression problems

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Cross-Entropy Loss: For classification problems.

$$\text{Cross-Entropy} = -\sum_{i=1}^{n} y_i \log(\hat{y}_i)$$

Backpropagation Algorithm

- **Purpose:** Adjusts the weights based on the error to minimize the loss function.

- **Steps:**
    1. **Calculate the loss:** Measure the error between predicted and actual values.
    2. **Compute gradients:** Use the chain rule to calculate the gradient of the loss with respect to each weight.
    3. **Update weights:** Adjust weights using gradient descent:

$$w := w - \eta \frac{\partial L}{\partial w}$$

-
    1. where η\etaη is the learning rate.

◆ 6. Types of Neural Networks

6.1 Feedforward Neural Networks (FNN)

- **Structure:** Data flows in one direction (input → output).
- **Use Cases:** Basic classification and regression tasks.

6.2 Convolutional Neural Networks (CNN)

- **Structure:** Specialized for image data. Applies convolutional filters to extract spatial features.
- **Components:**
    - Convolutional layers
    - Pooling layers
    - Fully connected layers
- **Applications:** Image recognition, object detection.

6.3 Recurrent Neural Networks (RNN)

- **Structure:** Designed for sequential data. Maintains a memory of previous inputs.
- **Variants:** Long Short-Term Memory (LSTM), Gated Recurrent Units (GRUs).
- **Applications:** Time series forecasting, natural language processing (NLP).

◆ 7. Challenges and Limitations

7.1 Overfitting

- **Problem:** Model performs well on training data but poorly on unseen data.
- **Solution:**
    - Regularization techniques (L2, Dropout).

- Early stopping.

7.2 Vanishing and Exploding Gradients

- **Problem:** Gradients become too small or too large during training, especially in deep networks.
- **Solution:**
  - Use ReLU activation.
  - Implement proper weight initialization techniques.

◆ 8. Real-World Applications in Machine Learning

1. **Computer Vision:**
   - **Face recognition:** Using CNNs to detect and classify faces.
   - **Object detection:** Autonomous vehicles, security cameras.
2. **Natural Language Processing (NLP):**
   - **Language translation:** Google Translate uses neural networks.
   - **Chatbots:** Use RNNs and transformers for conversational AI.
3. **Healthcare:**
   - **Disease diagnosis:** Predicting disease from medical images.
   - **Drug discovery:** Predicting molecular interactions.
4. **Finance:**
   - **Stock prediction:** Using RNNs to forecast stock trends.
   - **Fraud detection:** Identifying suspicious patterns in transactions.

Support Vector Machines (SVMs)

Definition:

Support Vector Machines (SVMs) are supervised learning models used primarily for classification and regression tasks. They work by finding the optimal hyperplane that best separates the data into distinct classes in high-dimensional space.

Key Concepts:

- **Hyperplane:** A decision boundary that separates different classes. In a 2D space, it's a line, and in 3D, it's a plane.
- **Support Vectors:** Data points that lie closest to the decision boundary. These points influence the position and orientation of the hyperplane.
- **Margin:** The distance between the hyperplane and the closest data points (support vectors) from each class. The goal of SVM is to maximize this margin.

Types of SVMs:

- **Linear SVM:** Works when the data is linearly separable.
- **Non-Linear SVM:** Utilizes the kernel trick to map data to higher-dimensional spaces for better classification of non-linearly separable data.

Kernel Functions:

- **Linear Kernel:** K(x,y)=x·yK(x, y) = x \cdot yK(x,y)=x·y
- **Polynomial Kernel:** K(x,y)=(x·y+c)dK(x, y) = (x \cdot y + c)^dK(x,y)=(x·y+c)d
- **Radial Basis Function (RBF) Kernel:** K(x,y)=exp(−γ‖x−y‖2)K(x, y) = \exp(-\gamma \| x - y \|^2)K(x,y)=exp(−γ‖x−y‖2)

Applications:

- Text classification (e.g., spam detection)
- Image classification
- Bioinformatics (e.g., protein classification)

---

2. Bayesian Networks

Definition:

Bayesian Networks (also known as Belief Networks) are probabilistic graphical models that represent a set of variables and their conditional dependencies via a directed acyclic graph (DAG).

Key Concepts:

- **Nodes:** Represent random variables (discrete or continuous).
- **Edges:** Directed links indicating conditional dependencies between variables.
- **Conditional Probability Table (CPT):** Defines the probability distribution for each node given its parent nodes.

Bayes' Theorem:

P(A|B)=P(B|A)·P(A)P(B)P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}P(A|B)=P(B)P(B|A)·P(A)

This theorem provides a way to update the probability of a hypothesis as more evidence is available.

Inference Types:

- **Predictive Inference:** Given known evidence, predict the probability of future outcomes.
- **Diagnostic Inference:** Determine the likelihood of causes based on observed outcomes.
- **Inter causal Reasoning:** Handle scenarios where multiple causes can explain a single effect.

Advantages:

- Handles uncertainty and incomplete data effectively.
- Incorporates prior knowledge into the learning process.
- Useful for causal reasoning and decision-making under uncertainty.

Applications:

- Medical diagnosis systems
- Spam filtering
- Fraud detection
- Risk assessment models

---

3. Bag-of-Words (BoW) Classifiers

Definition:

The Bag-of-Words (BoW) model is a representation of text data in natural language processing (NLP). It simplifies text by treating it as a collection (bag) of words, disregarding grammar and word order, but maintaining frequency information.

Steps to Build a BoW Classifier:

1. **Tokenization:** Break down the text into individual words (tokens).
2. **Vocabulary Creation:** Compile a list of all unique words (the vocabulary) in the dataset.
3. **Vectorization:** Represent each text as a vector indicating the frequency of each word in the vocabulary.

Example:

For sentences:

1. "The cat sat on the mat."
2. "The dog barked at the cat."

Vocabulary: `[the, cat, sat, on, mat, dog, barked, at]`

Vectors:

1. `[2, 1, 1, 1, 1, 0, 0, 0]`
2. `[2, 1, 0, 0, 0, 1, 1, 1]`

Limitations:

- Ignores word order and semantics.
- Large vocabulary size can increase model complexity and computational cost.

Enhancements:

- **TF-IDF (Term Frequency-Inverse Document Frequency):** Weighs words based on their importance across the corpus.
- **N-grams:** Incorporates phrases (e.g., bi-grams, tri-grams) to capture some contextual meaning.

Applications:

- Text classification (e.g., sentiment analysis)
- Document retrieval and ranking
- Spam detection

Detailed Overview: Machine Learning Classifiers

This note focuses on three key machine learning models: **Support Vector Machines (SVMs)**, **Bayesian Networks**, and the **Bag-of-Words (BoW) classifiers**. Each of these methods plays a vital role in different domains of machine learning, particularly in tasks like classification and pattern recognition.

---

1. Support Vector Machines (SVMs)

Definition:

Support Vector Machines (SVMs) are supervised learning models used primarily for classification and regression tasks. They work by finding the optimal hyperplane that best separates the data into distinct classes in high-dimensional space.

Key Concepts:

- **Hyperplane:** A decision boundary that separates different classes. In a 2D space, it's a line, and in 3D, it's a plane.
- **Support Vectors:** Data points that lie closest to the decision boundary. These points influence the position and orientation of the hyperplane.
- **Margin:** The distance between the hyperplane and the closest data points (support vectors) from each class. The goal of SVM is to maximize this margin.

Types of SVMs:

- **Linear SVM:** Works when the data is linearly separable.
- **Non-Linear SVM:** Utilizes the kernel trick to map data to higher-dimensional spaces for better classification of non-linearly separable data.

Kernel Functions:

- **Linear Kernel:** K(x,y)=x·yK(x, y) = x \cdot yK(x,y)=x·y
- **Polynomial Kernel:** K(x,y)=(x·y+c)dK(x, y) = (x \cdot y + c)^dK(x,y)=(x·y+c)d
- **Radial Basis Function (RBF) Kernel:** K(x,y)=exp⁡(−γ∥x−y∥2)K(x, y) = \exp(-\gamma \| x - y \|^2)K(x,y)=exp(−γ∥x−y∥2)

Applications:

- Text classification (e.g., spam detection)
- Image classification
- Bioinformatics (e.g., protein classification)

---

2. Bayesian Networks

Definition:

Bayesian Networks (also known as Belief Networks) are probabilistic graphical models that represent a set of variables and their conditional dependencies via a directed acyclic graph (DAG).

Key Concepts:

- **Nodes:** Represent random variables (discrete or continuous).
- **Edges:** Directed links indicating conditional dependencies between variables.
- **Conditional Probability Table (CPT):** Defines the probability distribution for each node given its parent nodes.

Bayes' Theorem:

P(A|B)=P(B|A)·P(A)P(B)P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}P(A|B)=P(B)P(B|A)·P(A)

This theorem provides a way to update the probability of a hypothesis as more evidence is available.

Inference Types:

- **Predictive Inference:** Given known evidence, predict the probability of future outcomes.
- **Diagnostic Inference:** Determine the likelihood of causes based on observed outcomes.
- **Intercausal Reasoning:** Handle scenarios where multiple causes can explain a single effect.

Advantages:

- Handles uncertainty and incomplete data effectively.
- Incorporates prior knowledge into the learning process.
- Useful for causal reasoning and decision-making under uncertainty.

Applications:

- Medical diagnosis systems
- Spam filtering
- Fraud detection
- Risk assessment models

---

3. Bag-of-Words (BoW) Classifiers

Definition:

The Bag-of-Words (BoW) model is a representation of text data in natural language processing (NLP). It simplifies text by treating it as a collection (bag) of words, disregarding grammar and word order, but maintaining frequency information.

Steps to Build a BoW Classifier:

1. **Tokenization:** Break down the text into individual words (tokens).
2. **Vocabulary Creation:** Compile a list of all unique words (the vocabulary) in the dataset.
3. **Vectorization:** Represent each text as a vector indicating the frequency of each word in the vocabulary.

Example:

For sentences:

1. "The cat sat on the mat."
2. "The dog barked at the cat."

Vocabulary: `[the, cat, sat, on, mat, dog, barked, at]`

Vectors:

1. `[2, 1, 1, 1, 1, 0, 0, 0]`
2. `[2, 1, 0, 0, 0, 1, 1, 1]`

Limitations:

- Ignores word order and semantics.
- Large vocabulary size can increase model complexity and computational cost.

Enhancements:

- **TF-IDF (Term Frequency-Inverse Document Frequency):** Weighs words based on their importance across the corpus.
- **N-grams:** Incorporates phrases (e.g., bi-grams, tri-grams) to capture some contextual meaning.

Applications:

- Text classification (e.g., sentiment analysis)
- Document retrieval and ranking
- Spam detection

Comparison and Use Cases:

Model	Type	Best For	Advantages
SVM	Supervised Classifier	High-dimensional data, binary classification	Effective in non-linear spaces with kernels
Bayesian Networks	Probabilistic Model	Uncertainty, causal relationships	Interpretable, incorporates prior knowledge
Bag-of-Words	Text Representation	Text classification and sentiment analysis	Simple, effective for large text datasets

N-gram Models

Definition:

An N-gram model is a probabilistic language model used to predict the next item in a sequence based on the previous N-1 items. It's widely used in natural language processing (NLP).

Key Concepts:

- **N-gram:** A contiguous sequence of N items (words, characters, etc.) from a given text or speech sample.
  - **Unigram:** Single word (N=1)
  - **Bigram:** Pair of words (N=2)
  - **Trigram:** Sequence of three words (N=3)

  Probability Calculation:

$$P(w_n|w_1, w_2, ..., w_{n-1}) \approx P(w_n|w_{n-N+1}, ..., w_{n-1})$$

- This simplifies the probability calculation by considering only the last N-1 words.

Smoothing Techniques:

- **Laplace Smoothing:** Adds a small value to all counts to handle zero probabilities.
- **Backoff and Interpolation:** Combines probabilities from higher and lower N-grams.

Applications:

- Predictive text and auto-completion
- Language modeling in speech recognition
- Text generation (e.g., chatbots)

---

2. Markov Models

Definition:

A Markov model is a stochastic model describing a sequence of events where the probability of each event depends only on the state of the previous event (Markov property).

Key Concepts:

- **Markov Property:** The future state depends only on the current state, not on the sequence of states that preceded it.

$$P(X_{n+1}|X_n, X_{n-1}, ..., X_1) = P(X_{n+1}|X_n)$$

- **State Transition Matrix:** Represents the probabilities of transitioning from one state to another.
- **Steady-State Distribution:** The long-term behavior of the system where the state probabilities remain constant.

Types:

- **First-order Markov Model:** Considers only the current state.
- **Higher-order Markov Model:** Considers multiple previous states.

Applications:

- Weather prediction
- Stock market modeling
- PageRank algorithm in web search

---

3. Hidden Markov Models (HMMs)

Definition:

A Hidden Markov Model is an extension of a Markov model where the system being

modeled is assumed to be a Markov process with hidden (unobserved) states. Observations depend on these hidden states through a probabilistic function.

Key Components:

- **States:** Hidden variables (not directly observable).
- **Observations:** Observable data influenced by the hidden states.
- **Transition Probabilities:** Probabilities of moving between hidden states.
- **Emission Probabilities:** Probabilities of observing a particular output from a hidden state.

Key Problems Solved by HMMs:

1. **Evaluation:** Given a sequence of observations, calculate the probability of the sequence.
2. **Decoding:** Determine the most likely sequence of hidden states for a given observation sequence (Viterbi algorithm).
3. **Learning:** Adjust the model parameters (Baum-Welch algorithm) to fit the observed data.

Applications:

- Speech recognition
- Part-of-speech tagging in NLP
- DNA sequence analysis

---

4. Probabilistic Relational Models (PRMs)

Definition:

Probabilistic Relational Models (PRMs) extend Bayesian networks to handle relational data, allowing probabilistic reasoning about entities and their relationships.

Key Concepts:

- **Relational Data:** Data that involves multiple objects with relationships between them, typically represented in databases.
- **Attributes:** Variables that describe the properties of objects.
- **Dependencies:** Capture probabilistic relationships not only between attributes of the same object but also across related objects.

Structure of PRMs:

- **Classes:** Define types of objects.
- **Attributes:** Variables associated with each class.
- **Relationships:** Probabilistic dependencies between attributes across related classes.

Advantages:

- Handles complex, structured data with rich interdependencies.
- Facilitates reasoning in domains where relationships are crucial (e.g., social networks).

Applications:

- Social network analysis
- Recommendation systems
- Knowledge representation in complex domains (e.g., biological networks)

Comparison and Use Cases:

Model	Type	Best For	Advantages
N-gram Models	Probabilistic Language Model	Sequence prediction in text data	Simple, effective for short-range dependencies
Markov Models	Stochastic Process Model	Predicting sequential events with simple state transitions	Efficient, interpretable
Hidden Markov Models (HMMs)	Probabilistic State Model	Complex sequences with hidden states (e.g., speech recognition)	Captures hidden patterns, flexible
Probabilistic Relational Models (PRMs)	Probabilistic Graphical Model	Complex relational data with dependencies	Handles structured, relational data

Association Rules

Definition:

Association rules are used to discover interesting relationships or patterns within large datasets. They are widely used in data mining, especially for market basket analysis.

Key Concepts:

- **Rule Structure:** An association rule is represented as X→YX, meaning "if X occurs, then Y is likely to occur."
  - **Antecedent (X):** Items on the left-hand side of the rule.
  - **Consequent (Y):** Items on the right-hand side of the rule.
- **Support:** Probability that both X and Y occur together

$$\text{Support}(X \to Y) = \frac{\text{Count}(X \cap Y)}{\text{Total Transactions}}$$

Confidence: Probability that Y occurs given that X has occurred.

$$\text{Confidence}(X \to Y) = \frac{\text{Count}(X \cap Y)}{\text{Count}(X)}$$

Lift: Measures how much more likely Y is to occur with X compared to random chance.

$$\text{Lift}(X \to Y) = \frac{\text{Confidence}(X \to Y)}{\text{Support}(Y)}$$

Algorithm:

- **Apriori Algorithm:** Identifies frequent itemsets and uses them to generate strong rules.

Applications:

- Market basket analysis (e.g., "customers who bought X also bought Y")
- Fraud detection
- Recommendation systems

---

2. Nearest Neighbor Classifiers (k-NN)

Definition:

Nearest Neighbor classifiers are a type of instance-based learning, where the class of a data point is determined by the majority class of its nearest neighbors in the feature space.

Key Concepts:

- **Distance Metrics:** Measure the similarity between data points. Common metrics include:
  - **Euclidean Distance:**

$$d(x, y) = \sqrt{\sum (x_i - y_i)^2}$$

Manhattan Distance: $$d(x, y) = \sum |x_i - y_i|$$

- **k (Number of Neighbors):** Determines how many nearest neighbors are used to make the classification. Choosing the optimal value of k is crucial for performance.

Algorithm Steps:

1. Calculate the distance between the new data point and all existing points.
2. Identify the k nearest neighbors.
3. Assign the class label based on a majority vote among these neighbors.

Advantages:

- Simple and easy to implement.
- No training phase; all computation occurs during prediction.

Disadvantages:

- Computationally intensive for large datasets.
- Sensitive to irrelevant or redundant features.

Applications:

- Text classification
- Image recognition
- Recommender systems

---

3. Locally Weighted Regression (LWR)

Definition:

Locally Weighted Regression (also called Local Regression or LOESS) is a non-parametric regression method where the model is fit locally around each data point, giving more weight to nearby points.

Key Concepts:

- **Local Fitting:** Fit a simple linear (or polynomial) regression model to a subset of the data near the point of interest.
- **Weighting Function:** Assigns weights to data points based on their distance from the target point. Common functions include:
    - **Gaussian Kernel:**

$$w_i = \exp\left(-\frac{d(x, x_i)^2}{2h^2}\right)$$

. **Tricube Kernel:**

$$w_i = (1 - |d(x, x_i)/h|^3)^3$$

- **Bandwidth (hhh):** Controls the size of the neighborhood and the smoothness of the regression curve.

Advantages:

- Flexible and adaptive to local data characteristics.
- Works well for non-linear relationships.

Disadvantages:

- Computationally expensive for large datasets.
- Sensitive to noise if bandwidth is too small.

Applications:

- Time series forecasting
- Non-linear regression problems
- Real-time control systems

---

4. Ensemble Classifiers

Definition:

Ensemble classifiers combine multiple base models (weak learners) to produce a more robust and accurate prediction than any single model.

Key Concepts:

- **Bagging (Bootstrap Aggregating):**
  - Creates multiple subsets of the training data using bootstrap sampling. Each subset is used to train a separate base model.
  - **Example:** Random Forest.
- **Boosting:**
  - Models are trained sequentially, and each new model corrects the errors made by the previous one.
  - **Example:** AdaBoost, Gradient Boosting, XGBoost.
- **Stacking:**
  - Combines predictions from multiple base models using another model (meta-learner) to make the final prediction.

Advantages:

- Improves accuracy and robustness.

- Reduces overfitting compared to single models.
- Handles complex decision boundaries.

Disadvantages:

- Increased computational cost.
- Interpretation can be difficult.

Applications:

- Image classification (e.g., face recognition)
- Fraud detection
- Financial forecasting

Comparison and Use Cases:

Model	Type	Best For	Advantages
Association Rules	Pattern Discovery	Discovering relationships in large datasets	Simple, interpretable, identifies patterns
Nearest Neighbor Classifiers	Instance-Based Classifier	Classification with small datasets or non-linear boundaries	Simple, no training phase, adaptable
Locally Weighted Regression	Non-Parametric Regression	Local, non-linear regression problems	Flexible, adaptive to local data patterns
Ensemble Classifiers	Combined Model Approach	Complex tasks with high variance or bias	High accuracy, reduces over fitting

KALINGA UNIVERSITY