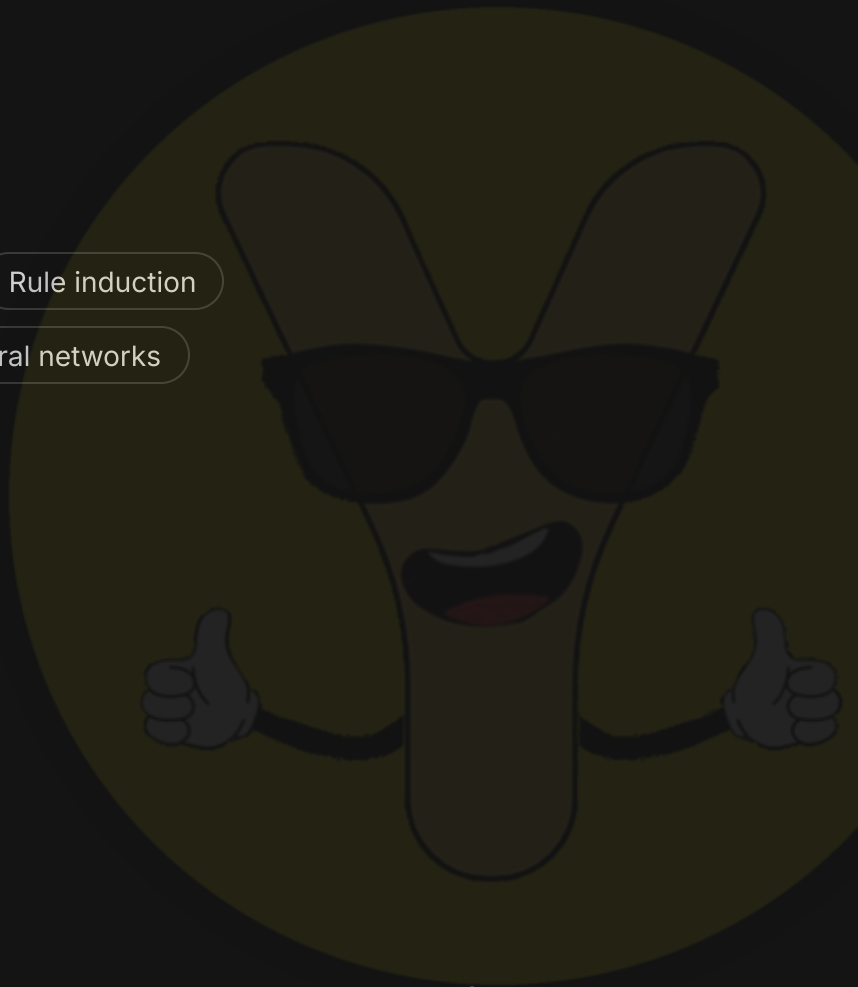
AKTU · 3RD YEAR · SEMESTER 6 · BADS601

Data Analytics

Unit 2 — Data Analysis

- Regression modeling
- Multivariate analysis
- Bayesian modeling, inference and Bayesian networks
- Support vector and kernel methods
- Analysis of time series: linear systems analysis & nonlinear dynamics
- Rule induction
- Neural networks: learning and generalisation
- Competitive learning
- Principal component analysis and neural networks
- Fuzzy logic: extracting fuzzy models from data
- Fuzzy decision trees
- Stochastic search methods



DECODED BY

Chirag Jain

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

Regression Modeling: Linear and Logistic

Regression modeling establishes statistical relationships between input features and target variables.

- Linear Regression predicts continuous targets using linear combinations: $y = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \epsilon$.
- Ordinary Least Squares (OLS) minimizes sum of squared residuals: $S_{SSE} = \sum_{i=1}^{m} (y_i - \hat{y}_i)^2$.
- Logistic Regression predicts binary probabilities using sigmoid link function: $P(Y = 1|X) = \frac{1}{1+e^{-(\beta_0+\beta^T \mathbf{x})}}$.
- Key assumptions: Linearity, independence of errors, homoscedasticity, and absence of multicollinearity.

Worked Example: Simple Linear Regression

Given data points $(1, 2)$, $(2, 4)$, and $(3, 5)$, find the optimal regression line parameters β_0 and β_1.

1. Calculate sample means: $\bar{x} = \frac{1+2+3}{3} = 2.0$, $\bar{y} = \frac{2+4+5}{3} = 3.67$.
2. Compute slope: $\beta_1 = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2} = \frac{(-1)(-1.67) + (0)(0.33) + (1)(1.33)}{(-1)^2 + 0^2 + 1^2} = \frac{3.0}{2} = 1.5$.
3. Compute intercept: $\beta_0 = \bar{y} - \beta_1 \bar{x} = 3.67 - 1.5(2.0) = 0.67$.

The estimated regression equation is $\hat{y} = 0.67 + 1.5x$.

Multivariate Analysis Techniques

Multivariate analysis simultaneously evaluates multiple variables to understand complex data structures.

- MANOVA (Multivariate ANOVA) tests mean vector differences across multiple dependent variables simultaneously.
- Factor Analysis identifies unobserved latent variables (factors) driving observed correlations.
- Canonical Correlation Analysis (CCA) measures linear relationships between two variable sets.
- Multivariate Normal Distribution forms the core baseline: $f(\mathbf{x}) = \frac{1}{(2\pi)^{p/2}|\mathbf{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mu)^T \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)}$.

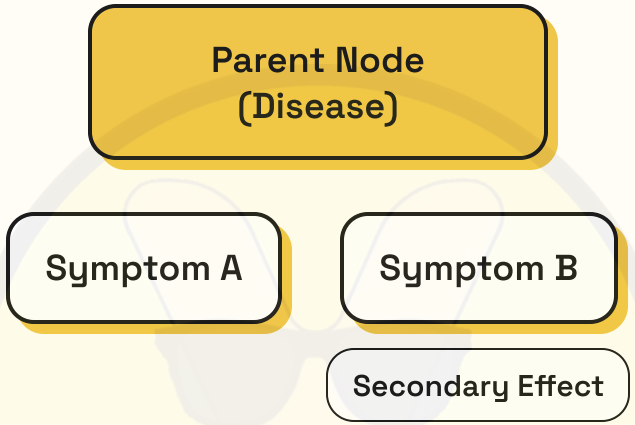
Bayesian Modeling and Inference

BAYESIAN INFERENCE

A statistical framework that updates the probability distribution of a hypothesis as new sample evidence is introduced.

- Bayes Theorem formulation: $P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$.
- Prior distribution $P(\theta)$ captures background knowledge before observing data D.
- Likelihood function $P(D|\theta)$ models data generation given parameter θ.
- Posterior distribution $P(\theta|D)$ represents updated belief post data observation.

Bayesian Networks Structure



Directed Acyclic Graph (DAG) representing conditional dependencies and factorization in a Bayesian Network.

Support Vector Machines and Kernel Methods

SVM constructs optimal hyperplanes in high-dimensional space for classification and regression tasks.

- Optimal Hyperplane maximizes the decision margin: $\text{Margin} = \frac{2}{\|\mathbf{w}\|}$.
- Primal Optimization Problem: $\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2$ subject to $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$.
- Kernel Trick maps data to higher dimensions implicitly: $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$.
- Popular Kernels: Radial Basis Function (RBF) $K(\mathbf{x}, \mathbf{z}) = e^{-\gamma\|\mathbf{x}-\mathbf{z}\|^2}$, Polynomial $K(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^T\mathbf{z} + c)^d$.

Linear Regression vs Support Vector Regression (SVR)

Parameter	Linear Regression	Support Vector Regression (SVR)
Optimization Goal	Minimizes total squared residual errors	Minimizes margin error bound within ϵ-tube
Sensitivity to Outliers	High (squared error penalties)	Low (robust due to ϵ-insensitive loss function)
Kernel Capability	Requires explicit feature engineering	Handles non-linear patterns via kernel functions
Computational Basis	Closed-form matrix inversion or GD	Convex Quadratic Programming (QP) optimization

Time Series: Linear Systems Analysis

Linear time series modeling analyzes stationary stochastic processes over sequential time intervals.

- Stationarity Requirement: Constant mean, variance, and autocovariance structure across time.
- Autoregressive AR(p) Model: $X_t = c + \sum_{i=1}^{p} \phi_i X_{t-i} + \epsilon_t$.
- Moving Average MA(q) Model: $X_t = \mu + \epsilon_t + \sum_{j=1}^{q} \theta_j \epsilon_{t-j}$.
- ARIMA(p, d, q) integrates differencing of degree d to achieve stationarity before ARMA fit.

Time Series: Nonlinear Dynamics

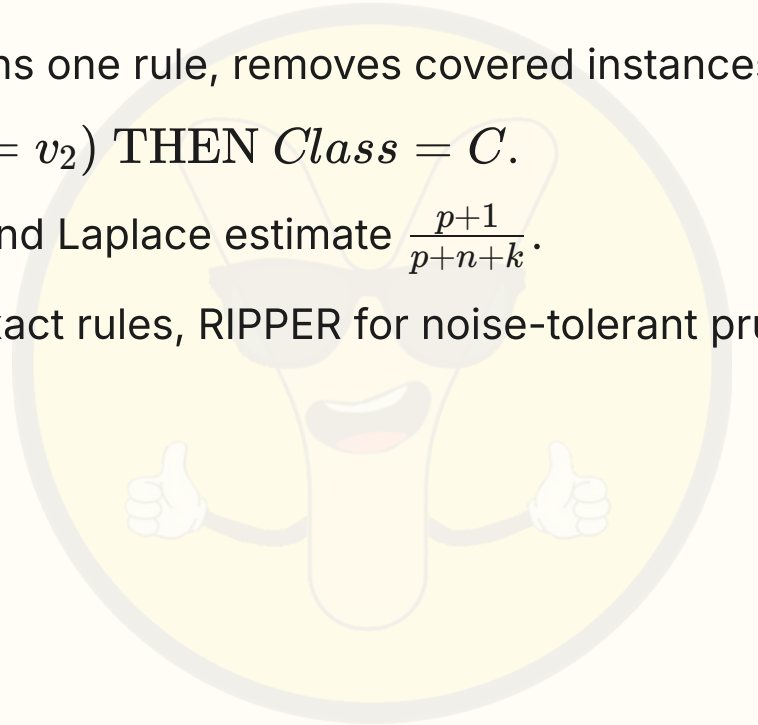
Nonlinear dynamics analyzes deterministic dynamic systems exhibiting non-periodic and chaotic behaviors.

- Phase Space Reconstruction maps scalar temporal observations into multidimensional state spaces using Takens' Delay Embedding.
- Attractors represent asymptotic state spaces (e.g., Strange Attractors in chaotic systems).
- Lyapunov Exponent λ measures exponential divergence rate of adjacent trajectories; $\lambda > 0$ signals chaos.
- Nonlinear modeling captures complex feedback mechanisms unresolvable by linear ARIMA models.

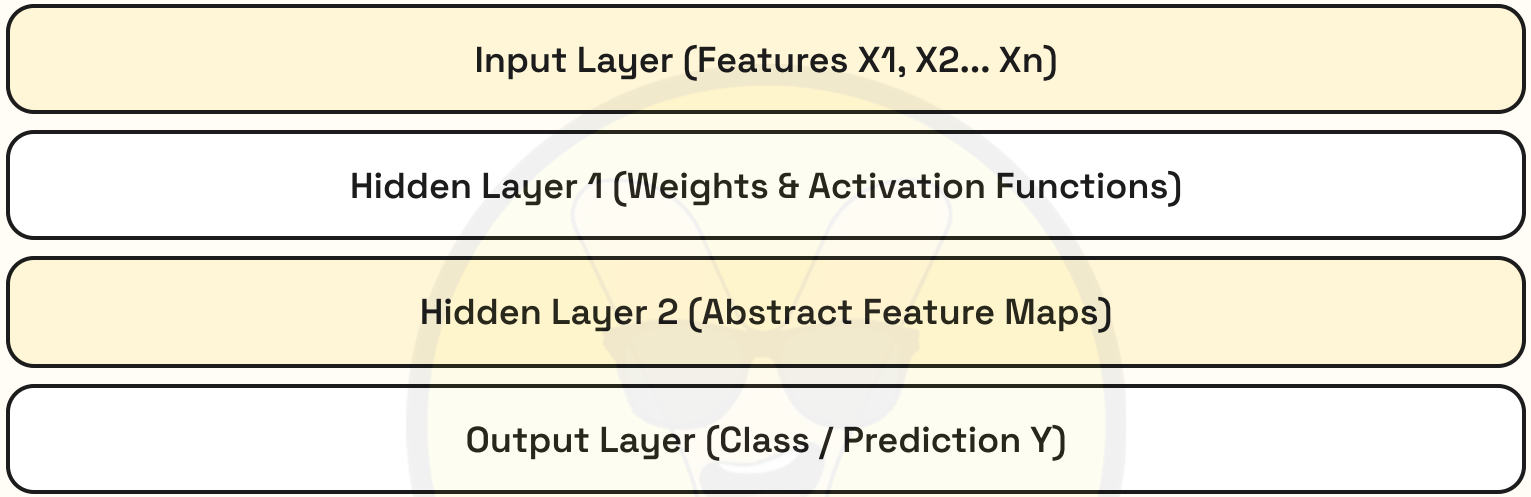
Rule Induction Algorithms

Rule induction extracts human-readable symbolic IF-THEN rules directly from observed data.

- Sequential Covering Strategy: Learns one rule, removes covered instances, and repeats recursively.
- Rule Format: $\text{IF } (A_1 = v_1) \wedge (A_2 = v_2) \text{ THEN } Class = C.$
- Evaluation Metrics: Precision $\frac{p}{p+n}$ and Laplace estimate $\frac{p+1}{p+n+k}$.
- Prominent Algorithms: PRISM for exact rules, RIPPER for noise-tolerant pruned rule sets.



Multilayer Perceptron Architecture

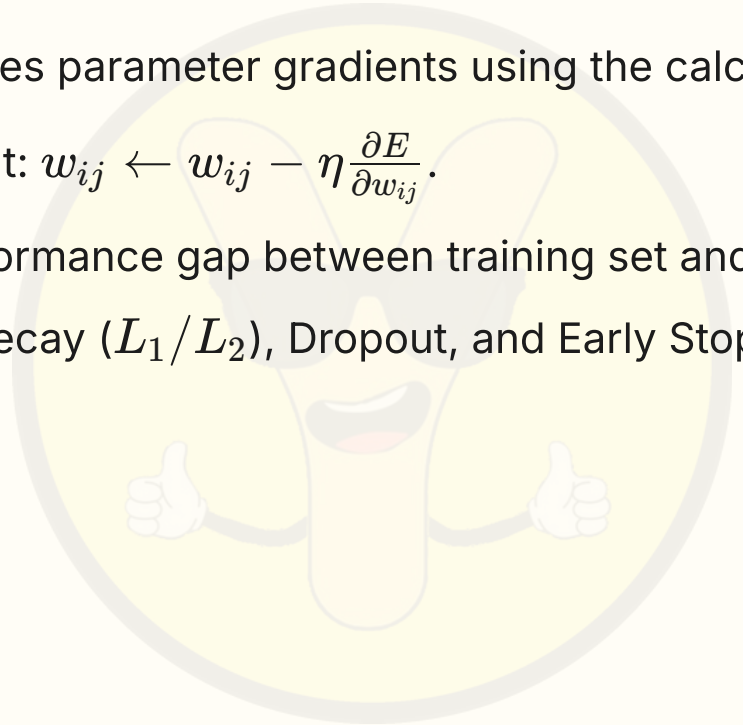


Standard feedforward neural network structure showing layer-wise forward propagation.

Neural Networks: Learning and Generalisation

Neural networks balance empirical training error minimization with generalisation to unseen samples.

- Backpropagation Algorithm computes parameter gradients using the calculus chain rule: $\frac{\partial E}{\partial w_{ij}} = \delta_j o_i$.
- Weight Update via Gradient Descent: $w_{ij} \leftarrow w_{ij} - \eta \frac{\partial E}{\partial w_{ij}}$.
- Generalisation Error measures performance gap between training set and independent test sets.
- Regularization Strategies: Weight decay (L_1/L_2), Dropout, and Early Stopping prevent overfitting.



Competitive Learning

COMPETITIVE LEARNING

An unsupervised neural network paradigm where output neurons compete to be activated, resulting in a single active neuron ('winner-take-all').

- Euclidean Distance Search: Winning neuron $k = \arg\min_i \|\mathbf{x} - \mathbf{w}_i\|$.
- Weight Update Rule for winner: $\mathbf{w}_k \leftarrow \mathbf{w}_k + \eta(\mathbf{x} - \mathbf{w}_k)$.
- Suppression: Non-winning neurons receive zero weight updates during the iteration.
- Primary architecture: Kohonen Self-Organizing Maps (SOM) for topological clustering.

PCA and Neural Network Realizations

Principal Component Analysis reduces feature dimensionality and can be trained via Hebbian neural learning.

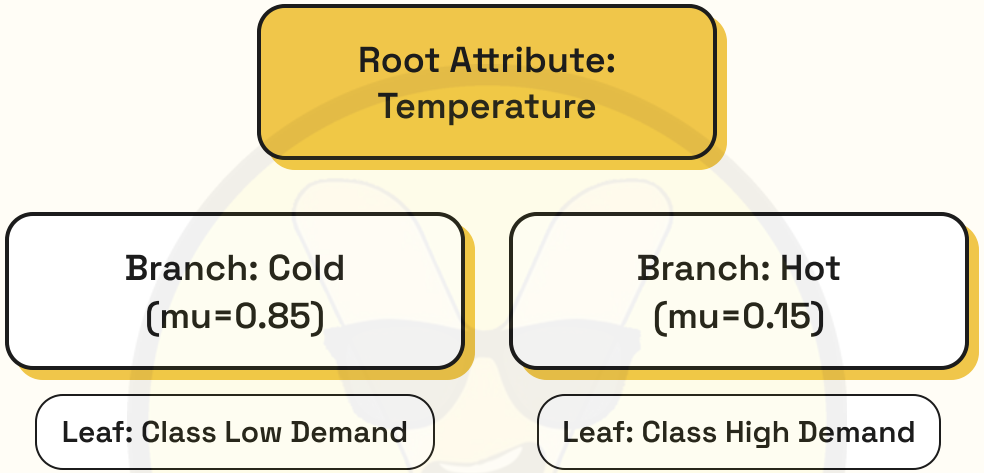
- Algebraic PCA finds eigenvectors of data covariance matrix $\mathbf{\Sigma} = \frac{1}{N}\mathbf{X}^T\mathbf{X}$.
- Oja's Learning Rule updates single neuron weights to extract first principal component: $\Delta\mathbf{w} = \eta y(\mathbf{x} - y\mathbf{w})$.
- Sanger's Rule (Generalized Hebbian Algorithm) extracts multiple orthogonal principal components: $\Delta\mathbf{w}_i = \eta y_i \left(\mathbf{x} - \sum_{k=1}^{i} y_k \mathbf{w}_k\right)$.
- Neural PCA avoids direct matrix decomposition for continuous high-dimensional streaming data.

Fuzzy Logic: Extracting Fuzzy Models from Data

Fuzzy logic models continuous relationships by translating quantitative numbers into degrees of membership.

- Fuzzification maps crisp input variables into fuzzy set degrees $\mu_A(x) \in [0, 1]$.
- Fuzzy Rules: $\text{IF } x \text{ is } A_i \text{ THEN } y \text{ is } B_i$.
- Data Extraction Methods: Fuzzy C-Means (FCM) clustering extracts rule antecedents from cluster centroids.
- Inference Engine Architectures: Mamdani (fuzzy linguistic outputs) vs. Takagi-Sugeno (functional linear outputs).

Fuzzy Decision Tree Structure



Soft partitioning in Fuzzy Decision Trees using continuous membership weights.

Stochastic Search Methods

Stochastic search algorithms explore complex non-convex search spaces using non-deterministic operations.

- Genetic Algorithms (GA): Mimic natural selection via selection, crossover, and mutation operators.
- Simulated Annealing (SA): Accepts worse solutions with probability $P = e^{-\Delta E/T}$ to escape local minima.
- Particle Swarm Optimization (PSO): Particles update velocity based on personal best p_{best} and global best g_{best}.
- Stochastic Gradient Descent (SGD): Introduces randomness by calculating gradients on mini-batch samples.

Deterministic Search vs Stochastic Search

Property	Deterministic Search	Stochastic Search
Execution Trajectory	Fixed and identical for same starting point	Randomized pathways across multiple runs
Local Minima Trap	High risk of getting stuck in local optima	Escapes local optima using probabilistic jumps
Computational Speed	Fast on simple convex functions	Slower but robust on complex non-convex spaces
Global Convergence	Guaranteed only for convex objective spaces	Probabilistic convergence to global optimum

Worked Example: Bayesian Inference Classification

A diagnostic test has prior $P(\text{Disease}) = 0.05$. Sensitivity $P(+|\text{Disease}) = 0.90$, and false positive rate $P(+|\text{No Disease}) = 0.10$. Calculate $P(\text{Disease}|+)$.

1. Calculate marginal probability $P(+) = P(+|\text{Disease})P(\text{Disease}) + P(+|\text{No Disease})P(\text{No Disease})$.
2. Compute numbers: $P(+) = (0.90 \times 0.05) + (0.10 \times 0.95) = 0.045 + 0.095 = 0.140$.
3. Apply Bayes Rule: $P(\text{Disease}|+) = \frac{P(+|\text{Disease})P(\text{Disease})}{P(+)} = \frac{0.045}{0.140} \approx 0.3214$.

The posterior probability of having the disease given a positive test result is 32.14%.

Exam Corner: High-Frequency AKTU Questions

Q1 Derive the soft-margin Support Vector Machine optimization problem and describe the function of kernel methods in handling non-linear data.

Q2 Explain the architecture, competitive learning mechanism, and weight update rule of Kohonen Self-Organizing Maps (SOM).

Q3 Differentiate between Mamdani and Takagi-Sugeno fuzzy inference systems. How are fuzzy models extracted from numerical data using clustering?

Q4 Formulate the mathematical structure of ARIMA(p,d,q) models for linear time series analysis. Contrast linear systems with non-linear dynamical systems.

Q5 Explain how neural networks extract principal components using Oja's rule and Sanger's Generalized Hebbian Algorithm.

Q6 Write short notes on: (a) Sequential Covering Algorithm for Rule Induction, (b) Particle Swarm Optimization (PSO), (c) Generalisation Error and Regularization.

Unit 2 done. You are exam-ready.

Revise the Exam corner once more, then open Unit 3 on theycotes.com. Every AKTU subject, unit by unit, free. Notes by Chirag Jain, Ycotes.

