

UNIT 3

Bayesian and Instance-Based Learning

(11 Hours) — Machine Learning (CACS456) · BCA IV/VIII

⚠ Please Note

These notes are prepared for quick study and **may contain errors or omissions**. Please cross-check formulas, definitions and worked examples with your textbook, class notes and teacher before relying on them in exams.

3.1 Probability Theory and Bayes Rule

Probabilistic learning treats uncertainty explicitly: instead of a hard answer, the model outputs how likely each hypothesis or class is given the data.

Concept	Definition / Formula
Probability $P(A)$	Likelihood of event A , $0 \leq P(A) \leq 1$; $P(\bar{A}) = 1 - P(A)$
Joint probability	$P(A, B) = P(A \cap B)$ — both occur
Conditional probability	$P(A B) = P(A, B) / P(B)$, for $P(B) > 0$
Product rule	$P(A, B) = P(A B) P(B) = P(B A) P(A)$
Sum / total probability	$P(B) = \sum_i P(B A_i) P(A_i)$ for mutually exclusive, exhaustive A_i
Independence	A, B independent $\Leftrightarrow P(A, B) = P(A)P(B) \Leftrightarrow P(A B) = P(A)$
Conditional independence	$A \perp B C \Leftrightarrow P(A, B C) = P(A C) P(B C)$
Expectation	$E[X] = \sum x \cdot P(x)$

Bayes Rule

Bayes' Theorem

$$P(h | D) = P(D | h) \cdot P(h) / P(D)$$

Posterior $P(h | D)$: belief in hypothesis h after seeing data D .

Likelihood $P(D | h)$: how probable the data are if h is true.

Prior $P(h)$: belief before seeing data. **Evidence** $P(D)$: normalising constant $= \sum_h P(D | h)P(h)$.

Example 3.1 — Medical test

A disease affects 1% of people. A test detects it 95% of the time when present (sensitivity), and gives a false positive 5% of the time when absent. A person tests positive. What is the probability they have the disease?

Solution

$$P(D) = 0.01, P(\neg D) = 0.99, P(+ | D) = 0.95, P(+ | \neg D) = 0.05$$

$$P(+) = 0.95 \times 0.01 + 0.05 \times 0.99 = 0.0095 + 0.0495 = 0.059$$

$$P(D | +) = 0.0095 / 0.059 \approx \mathbf{0.161 \text{ (16.1\%)}}$$

Despite a positive test the chance is only ~16% because the disease is rare (low prior).

MAP vs ML hypothesis: the *Maximum A Posteriori* hypothesis $h_{\text{MAP}} = \operatorname{argmax} P(D | h)P(h)$. If all priors are equal this reduces to the *Maximum Likelihood* hypothesis $\operatorname{argmax} P(D | h)$.

3.2 Classifying with Bayes Decision Theory

Given a feature vector x , Bayes decision theory assigns x to the class with the highest posterior probability. This rule minimises the probability of misclassification (the **Bayes error**, the lowest possible error for the problem).

Bayes Decision Rule

Choose class ω_i if $P(\omega_i | x) > P(\omega_j | x)$ for all $j \neq i$

Since $P(\omega_i | x) = P(x | \omega_i)P(\omega_i)/P(x)$ and $P(x)$ is common: choose **$\operatorname{argmax} P(x | \omega_i) P(\omega_i)$**

Two classes: decide ω_1 if the *likelihood ratio* $P(x | \omega_1)/P(x | \omega_2) > P(\omega_2)/P(\omega_1)$.

Minimum-risk form: if misclassification has different costs, define loss $\lambda(\alpha_i | \omega_j)$ for taking action α_i when the true class is ω_j . The conditional risk is $R(\alpha_i | x) = \sum_j \lambda(\alpha_i | \omega_j) P(\omega_j | x)$, and we choose the action with minimum risk. With 0–1 loss this equals picking the maximum posterior.

Naïve Bayes Classifier

When $x = (x_1, \dots, x_n)$ has many features, estimating $P(x | \omega)$ is hard. **Naïve Bayes** assumes features are conditionally independent given the class:

Naïve Bayes

$P(\omega | x_1 \dots x_n) \propto P(\omega) \cdot \prod P(x_j | \omega) \rightarrow \text{predict argmax over classes}$

Estimate priors and likelihoods by counting in training data. Use **Laplace smoothing** (add 1 to counts) to avoid zero probabilities.

Example 3.2 — Play Tennis (simplified)

Training data (10 days): Play = Yes in 6 days, No in 4 days.

Among Yes days: Sunny in 2, Wind Weak in 4. Among No days: Sunny in 3, Wind Weak in 1.

Classify a new day: Outlook = Sunny, Wind = Weak.

Solution

Yes: $P(\text{Yes}) \cdot P(\text{Sunny}|\text{Yes}) \cdot P(\text{Weak}|\text{Yes}) = (6/10)(2/6)(4/6) = 0.6 \times 0.333 \times 0.667 = \mathbf{0.1333}$

No: $P(\text{No}) \cdot P(\text{Sunny}|\text{No}) \cdot P(\text{Weak}|\text{No}) = (4/10)(3/4)(1/4) = 0.4 \times 0.75 \times 0.25 = \mathbf{0.075}$

Normalised: $P(\text{Yes} | x) = 0.1333 / (0.1333 + 0.075) = \mathbf{0.64} \rightarrow \text{predict Play = Yes.}$

- **Pros:** fast, works with small data, handles high dimensions (text classification, spam filtering).
- **Cons:** independence assumption rarely true; zero-frequency problem; poor probability estimates.

Variant	Feature type
Gaussian NB	Continuous features assumed normally distributed per class
Multinomial NB	Counts (word frequencies in documents)
Bernoulli NB	Binary features (word present / absent)

3.3 Conditional Probability — Worked Example

Example 3.3

A box has 3 red and 2 blue balls. Two balls are drawn without replacement. Find $P(\text{second is red} | \text{first is red})$ and $P(\text{both red})$.

Solution

After removing one red: 2 red, 2 blue remain $\rightarrow P(R_2 | R_1) = 2/4 = \mathbf{0.5}$

$P(R_1) = 3/5 = 0.6 \rightarrow P(R_1, R_2) = P(R_2 | R_1) \cdot P(R_1) = 0.5 \times 0.6 = \mathbf{0.3}$

3.4 Bayesian Belief Network (BBN)

A Bayesian Belief Network (Bayesian network) is a **directed acyclic graph (DAG)** that represents a joint probability distribution compactly. **Nodes** are random variables; a directed **edge** $A \rightarrow B$ means A directly influences B . Each node has a **conditional probability table (CPT)** giving $P(\text{node} \mid \text{parents})$. Absence of an edge encodes conditional independence.

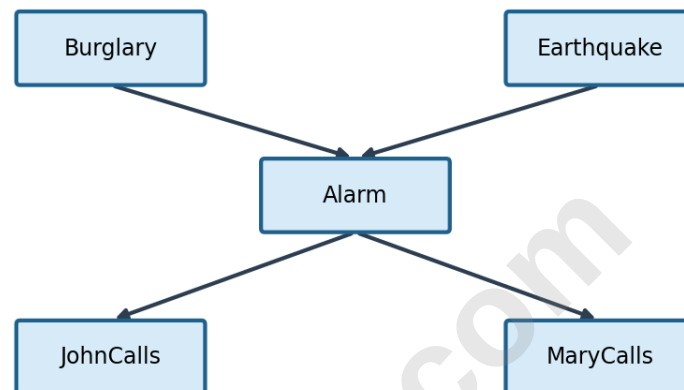


Figure 3.1: Classic alarm network: Burglary and Earthquake cause the Alarm, which makes John and Mary call.

Joint Distribution Factorisation

$$P(X_1, \dots, X_n) = \prod P(X_i \mid \text{Parents}(X_i))$$

Alarm network: $P(B, E, A, J, M) = P(B) \cdot P(E) \cdot P(A \mid B, E) \cdot P(J \mid A) \cdot P(M \mid A)$

This needs far fewer parameters than the full joint table ($2^5 - 1 = 31$ vs 10 here).

Example 3.4

$P(B) = 0.001$, $P(E) = 0.002$, $P(A \mid B, \neg E) = 0.94$, $P(J \mid A) = 0.90$, $P(M \mid A) = 0.70$. Find $P(B, \neg E, A, J, M)$.

Solution

$$= P(B) \cdot P(\neg E) \cdot P(A \mid B, \neg E) \cdot P(J \mid A) \cdot P(M \mid A)$$

$$P(\neg E) = 1 - 0.002 = 0.998$$

$$= 0.001 \times 0.998 \times 0.94 \times 0.90 \times 0.70 \approx \mathbf{5.9 \times 10^{-4}}$$

- **Inference:** compute posteriors such as $P(\text{Burglary} \mid \text{JohnCalls}, \text{MaryCalls})$ by enumeration, variable elimination or sampling.

- **Learning:** parameters from data by counting (ML) or EM when variables are hidden; structure by score-based or constraint-based search.
- **Naïve Bayes** is the special case where the class node is the single parent of every feature.

3.5 K-Nearest Neighbour (k-NN)

k-NN is an **instance-based (memory-based, lazy)** learner. It does no explicit training: it stores all training examples, and at prediction time finds the k closest to the query point and lets them vote (classification) or averages them (regression). It is **non-parametric** because it makes no assumption about the data distribution.

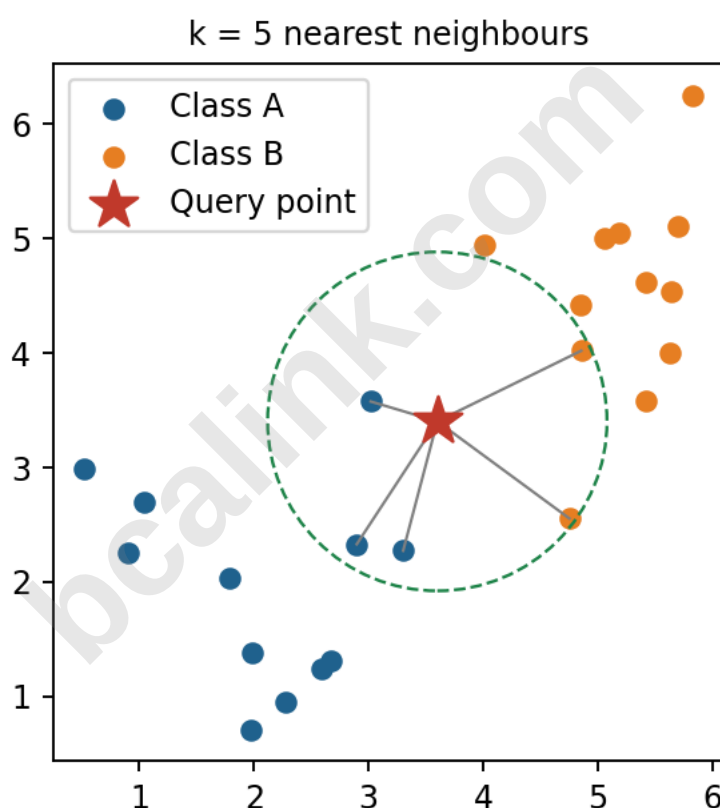


Figure 3.2: k-NN with $k = 5$ — the query point takes the majority class of its five nearest neighbours.

Algorithm

1. Choose k and a distance metric.
2. Compute the distance from the query x to every training point.
3. Select the k nearest points.
4. Classification: majority vote (optionally weighted by $1/d^2$). Regression: mean of neighbours' values.

Distance Measures

Euclidean: $d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum (\mathbf{x}_i - \mathbf{y}_i)^2}$ Manhattan: $\sum |\mathbf{x}_i - \mathbf{y}_i|$

Minkowski: $(\sum |\mathbf{x}_i - \mathbf{y}_i|^p)^{1/p}$ Hamming: number of differing attributes (categorical data)

Normalise features (min-max or z-score) first, otherwise large-scale features dominate.

Example 3.5

Training data (Height cm, Weight kg $\rightarrow$ Class): A(170, 65 $\rightarrow$ Normal), B(180, 90 $\rightarrow$ Over), C(160, 50 $\rightarrow$ Under), D(175, 80 $\rightarrow$ Over). Classify query (172, 70) with $k = 3$ using Euclidean distance.

Solution

$$d(Q, A) = \sqrt{(2^2 + 5^2)} = \sqrt{29} \approx 5.39$$

$$d(Q, B) = \sqrt{(8^2 + 20^2)} = \sqrt{464} \approx 21.54$$

$$d(Q, C) = \sqrt{(12^2 + 20^2)} = \sqrt{544} \approx 23.32$$

$$d(Q, D) = \sqrt{(3^2 + 10^2)} = \sqrt{109} \approx 10.44$$

3 nearest: A (Normal), D (Over), B (Over) $\rightarrow$ majority = **Over**

Choice of k	Effect
Small k (e.g. 1)	Low bias, high variance — sensitive to noise, overfits
Large k	High bias, low variance — smoother boundary, may underfit
Practical rule	Pick odd k (avoid ties) using cross-validation; $k \approx \sqrt{N}$ is a starting point

- **Pros:** simple, no training time, naturally multi-class, adapts instantly to new data.
- **Cons:** slow prediction on large data ($O(N \cdot d)$), memory heavy, suffers from the **curse of dimensionality**, sensitive to irrelevant features and scaling. Speed-ups: KD-trees, ball trees, approximate NN.

Key Takeaways — Unit 3

- ✓ Bayes rule: posterior = likelihood $\times$ prior / evidence.
- ✓ Bayes decision rule: pick the class with maximum posterior; it minimises error.
- ✓ Naïve Bayes assumes conditional independence of features: $P(\omega) \cdot \prod P(x_j | \omega)$.
- ✓ Bayesian networks are DAGs with CPTs; joint = $\prod P(X_i | \text{Parents})$.
- ✓ k-NN is a lazy, instance-based method: vote among the k nearest points; scale features and tune k.
- ✓ Small k $\rightarrow$ overfit; large k $\rightarrow$ underfit.

— ***End of Unit 3*** —

These notes may contain errors — please verify with your textbook and teacher.

bcalink.com