Knowledge Representation and Reasoning

- Intro Lab -

Naive Bayes Classification

Naive Baye Classification – the definition

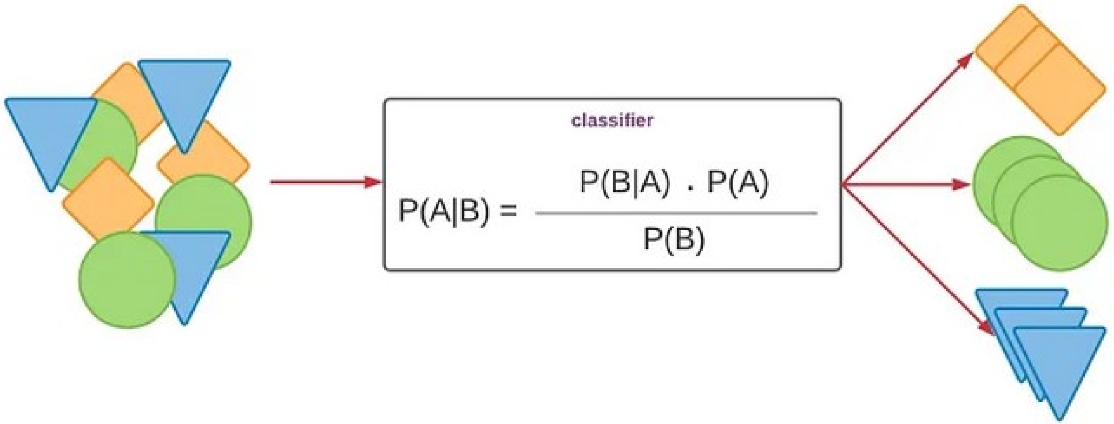
The Naive Bayes (NB) prediction algorithm is a **non-parametric probabilistic classifier**, based on the application of **Bayes' Theorem** with a strong (naive) **assumption** on the **independence of features.**

Key aspects:

- Knowledge in a NB algorithm is represented as **probabilities.**
- The **probabilities** capture **only** the relationship between the **target class** and **each individual feature** (attributes in the data)
- Non-parametric → there are no parameters to train; we care about **estimating** the probabilities of the dependence of a feature (attribute) on the target class **using all avaialable data**

Naive Baye Classification – breaking it down

The Naive Bayes (NB) algorithm is **probabilistic classifier.**



Example:

- **Class: shape type**
- **Features (attributes)** of the data: number of straight line segments, color, permiter length

Bayes Theorem

Likelihood of the Evidence given that the Hypothesis is True

Prior Probability of the Hypothesis

$$P(H|E) = \frac{P(E|H) * P(H)}{P(E)}$$

Posterior Probability of the Hypothesis given that the Evidence is True

Prior Probability that the evidence is True

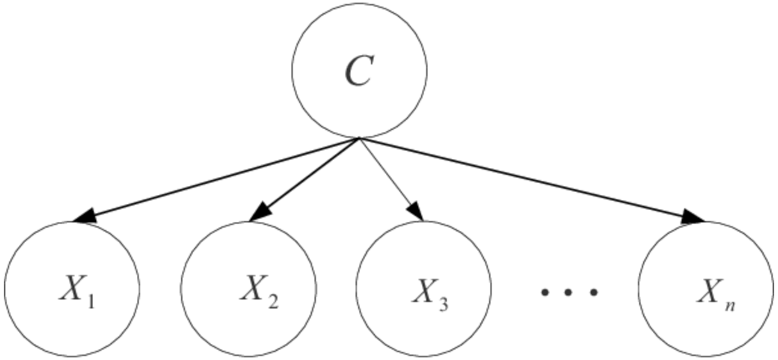
In classification:

- Hypotheses → which class?
- Evidence → features of the data sample

Naive Baye Classification – breaking it down

Knowledge representation assumption:

- **independence of features (naive)**
- **Conditional** dependence of each feature on the class



Likelihood

Class Prior Probability

$$P(c \mid x) = \frac{P(x \mid c)\,P(c)}{P(x)}$$

Posterior Probability

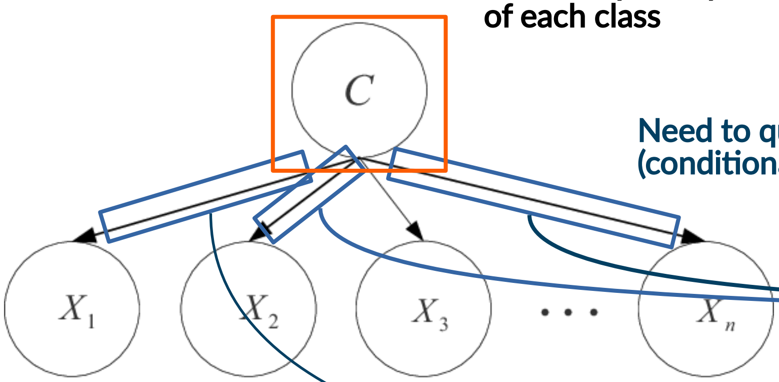
Predictor Prior Probability

$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Knowledge representation assumption:

- **independence of features (naive)**
- **Conditional** dependence of **each feature** on **the class**

Need to quantify the prior probability of each class



Need to quantify these dependencies (conditional probabilities)

Likelihood

Class Prior Probability

$$P(c\,|\,x) = \frac{P(x\,|\,c)\,P(c)}{P(x)}$$

Posterior Probability

Predictor Prior Probability

$$P(c\,|\,X) = \boxed{P(x_1\,|\,c)} \times \boxed{P(x_2\,|\,c)} \times \cdots \times \boxed{P(x_n\,|\,c)} \times \boxed{P(c)}$$

The algorithmic approach for the case of discrete events, i.e. **estimating probabilities by counting – frequency of apparition (class)** and **co-apparation (feature – class)**

$$P(C=c)=\frac{\text{\# samples in class c}}{\text{\# total number of samples}}$$

Laplace smoothing

$$P(X_i=x_i|C=c)=\frac{\text{\# apparitions of feature } x_i \text{ in samples of class } c+\boxed{\alpha}}{\text{\# total number of values of feature } X_i \text{ in class } c+\boxed{\text{\# num unique values of feature } X_i\cdot\alpha}}$$

$$C_{NB}=\arg\max_{c\in C}\left[p(C=c)\cdot\prod_i^N P(X_i=x_i|C=c)\right]$$

Naive Baye Classification – breaking it down

The algorithmic approach for the case of discrete events, i.e. **estimating probabilities by counting – frequency of apparition (class)** and **co-apparation (feature – class)**

$$C_{NB} = arg\,max_{c \in C} \left[p(C=c) \cdot \prod_{i}^{N} P(X_i = x_i | C=c) \right]$$

Product of probabilities becomes numerically unstable when multiplying lots of them => apply **log()**

$$C_{NB} = arg\,max_{c \in C} \left[\log(p(C=c)) + \sum_{i}^{N} \log(P(X_i = x_i | C=c)) \right]$$

Naive Baye Classification – the task

Apply Naive Bayes Classification to the **Nursery** dataset.

Features

parents: usual, pretentious, great_pret

has_nurs: proper, less_proper, improper, critical, very_crit

form: complete, completed, incomplete, foster

children: 1, 2, 3, more

housing: convenient, less_conv, critical

finance: convenient, inconv

social: nonprob, slightly_prob, problematic

health: recommended, priority, not_recom

Class values

not_recom, recommend, very_recom, priority, spec_prior

- Split data into 80% train, 20% test using **scikit-learn** train_test_split() method. Use a seed for the random number generator.
- Compute:
  - General Accuracy
  - Precision, Recall, F1 score table for each class value
  - Print a confusion matrix