

## 1 Bayesian Classifiers

### Probabilities

#### Sum Rule

$$P(A) = \sum_B P(A, B)$$

$$P(A) = \sum_B \sum_C P(A, B, C)$$

#### Product Rule

$$P(A, B) = P(B|A) \times P(A) = P(A|B) \times P(B)$$

### Bayes Theorem

$$P(A|B) = \frac{P(A, B)}{P(B)} = \frac{P(B|A)P(A)}{P(B)}$$

(Generalised case)

$$P(A_1 \dots A_k | B_1 \dots B_p) = \frac{P(B_1 \dots B_p, A_1 \dots A_k)}{P(B_1 \dots B_p)}$$

### Bayesian Classifiers

Bayesian classifiers aim to find the mapping  $f: \mathbf{x} \Rightarrow y$  for supervised learning in the form of conditional probability  $P(y|\mathbf{X})$  via Bayes rule.

$$P(y|\mathbf{X}) = \frac{P(y, \mathbf{X})}{P(\mathbf{X})} = \frac{P(\mathbf{X}|y)P(y)}{P(\mathbf{X})}$$

For a classification with C classes, given a data instance  $\mathbf{x}^*$ :

$$y^* = c^* \text{ if } c^* = \underset{c}{\operatorname{argmax}} P(y = c | \mathbf{x}^*)$$

Applying Bayes rule,

$$P(y = c | \mathbf{x}^*) = \frac{P(\mathbf{x}^* | y = c)P(y = c)}{P(\mathbf{x}^*)}$$

Therefore,

$$y^* = \underset{c}{\operatorname{argmax}} \frac{P(\mathbf{x}^* | y = c)P(y = c)}{P(\mathbf{x}^*)}$$

$$= \underset{c}{\operatorname{argmax}} P(\mathbf{x}^* | y = c)P(y = c)$$

## 2 Bayesian Decision Theory

Incorporating cost of misclassification on top of simple Bayesian Classifiers.

### Loss/Cost

Actions:  $a_c$ , i.e., predict  $y = c$

Define  $\lambda_{ij}$  as the cost of  $a_i$  when optimal action is  $a_j$ . E.g.:

$$\lambda_{00} = 0 \text{ (predict correctly)}$$

$$\lambda_{11} = 0 \text{ (predict correctly)}$$

$$\lambda_{01} = 10 \text{ misclassify 1 as 0}$$

$$\lambda_{00} = 1 \text{ misclassify 0 as 1}$$

### Expected Risk

Expected risk for taking action  $a_i$ :

$$R(a_i | \mathbf{x}) = \sum_{c=0}^{C-1} \lambda_{ic} P(y = c | \mathbf{x})$$

To classify, for all actions, calculate expected risk, then choose the action with the minimum risk.

**Special Case: 0/1 loss**

$$\lambda_{ij} = \begin{cases} 0 & \text{if } i = j \\ 1 & \text{if } i \neq j \end{cases}$$

$$\therefore R(a_i | \mathbf{x}) = 1 - P(y = i | \mathbf{x})$$

In this case,

$$\text{Choose } a_i \text{ if } R(a_i | \mathbf{x}) = \min_{a_c} R(a_c | \mathbf{x})$$

Is equivalent to:

$$\text{Predict } y = c^*$$

$$\text{if } P(y = c^* | \mathbf{x}) = \max_c P(y = c | \mathbf{x})$$

## 3 Naïve Bayes Classifiers

### Independence

A is **independent** of B, if:

$$P(A, B) = P(A|B) \times P(B) = P(A) \times P(B)$$

$$P(A, B) = P(B|A) \times P(A) = P(A) \times P(B)$$

Or,

$$P(A|B) = P(A)$$

$$P(B|A) = P(B)$$

### Conditional Independence

A is **conditionally independent** of B, given C if:

$$P(A|B, C) = P(A|C)$$

### Naïve Bayes Classifier

1. Assumption: conditional independence of features given label

$$p(\mathbf{x} | y = c) = P(x_1, \dots, x_d | y = c)$$

$$= P(x_1 | y = c)P(x_2 | y = c) \dots P(x_d | y = c)$$

$$= \prod_{i=1}^d P(x_i | y = c)$$

To classify a test record  $\mathbf{x}^*$ , compute the posteriors for each class:

$$p(y = c | \mathbf{x}^*) = \frac{(\prod_{i=1}^d P(x_i^* | y = c))P(y = c)}{P(\mathbf{x}^*)}$$

Since  $P(\mathbf{x}^*)$  is constant for each class  $c$ , it is sufficient to choose the class that maximises the numerator term.

$$y^* = \underset{c}{\operatorname{argmax}} \left( \prod_{i=1}^d P(x_i^* | y = c) \right) P(y = c)$$

### Estimating Cond Prob (Discrete)

$$P(x_i = k | y = c) = \frac{|(x_i - k) \wedge (y = c)|}{|y = c|}$$

### Estimating Cond Prob (Continuous)

$$P(x_i | y = c) = \frac{1}{\sqrt{2\pi\sigma_{ic}^2}} e^{-\frac{(x_i - \mu_{ic})^2}{2\sigma_{ic}^2}}$$

Supposing there are  $N_c$  instances in class  $c$ ,  
Sample mean:

$$\mu_{ic} = \frac{1}{N_c} \sum_{j=1}^{N_c} x_{ij}$$

Sample variance:

$$\sigma_{ic}^2 = \frac{1}{N_c - 1} \sum_{j=1}^{N_c} (x_{ij} - \mu_{ic})^2$$

### Laplace Estimate

Alternative prob estimation for discrete features.

$$P(x_i = k | y = c) = \frac{|(x_i - k) \wedge (y = c)| + 1}{|y = c| + n_i}$$

where  $n_i$  is #distinct values of  $x_i$ . In extreme cases with no training data,  
 $P(x_i = k | y = c) = \frac{1}{n_i}$

### M-estimate

A more general estimation:

$$P(x_i = k | y = c) = \frac{|(x_i - k) \wedge (y = c)| + m\bar{P}}{|y = c| + m}$$

Where  $m$  is a hyperparameter and  $\bar{P}$  is prior information of  $P(x_i = k | y = c)$ . (e.g., domain knowledge)

Extreme case with no training data:

$$P(x_i = k | y = c) = \bar{P}(x_i = k | y = c)$$

## 4 Bayesian Belief Networks

Suppose all features are **discrete** (if there are continuous and discrete, estimation is much more difficult)

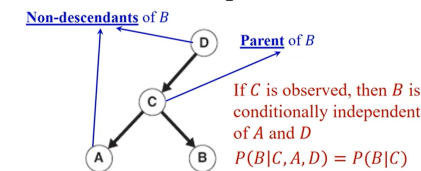
Two key elements:

1. A directed acyclic graph (DAG) encoding dependence relationships between a set of variables

2. A probability table associating each node to immediate parent nodes

### DAG: Conditional Independence

A node in a Bayesian network is conditionally independent of its non-descendants, **if its parents are known**.



**IMPORTANT!** If A and B are conditionally independent given C, we have:

1.  $P(A|B, C) = P(A|C)$
2.  $P(A, B|C) = P(A|C)P(B|C)$

### Important! Using BBN for Inference

Given a BBN, and an inference(prediction) task:

1. Translate problem into probabilistic language

2. If the probabilities to be estimated cannot be obtained from the probability tables of the BBN, then

A. Identify a subgraph which captures the dependence between input variables (features) and output variable (class)

B. Based on the network topology, apply product rule, sum rule and the properties of conditional independence and independence to induce equivalent forms of the probabilities until all probabilities can be found from the probability tables.