Bayesians

Bayesian Learning is a method of **statistical inference** in which **Bayes' theorem** is used to **update the probability** for a hypothesis as more evidence or information becomes available.

The **Bayes' theorem** is:

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

- A is the **hypothesis** and B is the **evidence**;
- Bayesian classifiers choose the **most probable class** given the evidence (data training).

MAP Classifier

- The **MAP (Maximum A Posteriori) Classifier** is a Bayesian classifier that uses the **maximum a posteriori** decision rule to classify a new object:

$$x \to y$$

$$\hat{y} = \underset{c_i \in C}{\text{argmax}} P(c_i|x) = \underset{c_i \in C}{\text{argmax}} \frac{P(x|c_i)P(c_i)}{P(x)}$$

The MAP classifier is:

$$\hat{y} = \underset{c_i \in C}{\text{argmax}} P(c_i) \prod_{j=1}^{d} P(a_j|c_i)$$

In Bayesian classifiers:

- Records are represented as **tuples** of d values;
- **Training algorithm** - to compute **prior probabilities** for each class;

- **Classification procedure -** to estimate likelihood for Z given each class, and then to classify Z as the most probable class;
- In the case of **equi-probable classes**, the classifier is not able to distinguish between them;
- **Estimation of prior probabilities** - the probability of each class is estimated by the **relative frequency** of the class in the training set:

$$P(c_i) = \frac{\mathrm{n}_i}{\mathrm{n}}$$

- **Estimation of likelihood -** the probability of each attribute value given the class is estimated by the **relative frequency** of the attribute value in the class:

$$P(x|c_i) = \frac{\mathrm{n}_{x|i}}{\mathrm{n}_i}$$

- n_i is the number of records in the class c_i;
- n is the total number of records;
- $n_{x|i}$ is the number of records in the class c_i with the attribute value x.

If we use numeric variables, we can use **probability density functions** to estimate the likelihood:

$$P(x|c_i) = f_i(x|\mu_i, \sigma_i)$$

$$X_i \sim N(\mu_i, \sigma_i^2)$$

Finally, if there are multiple variables, we need to **jointly estimate** the likelihood:

$$\vec{X} \sim N(\vec{\mu}, \Sigma^2)$$

$$P(\vec{x}|c_i) = f_i(\vec{x}|\vec{\mu}_i, \Sigma_i^2)$$

Naive Bayes Algorithm

Naive Bayes Assumption: all variables are **conditionally independent** given the class.

$$\hat{y} = \underset{c_i \in C}{\text{argmax}} P(c_i) \prod_{j=1}^{d} P(a_j | c_i)$$

- **Training algorithm -** to compute **prior probabilities** for each class;
- **Classification procedure -** to estimate likelihood for Z **individual dimensions given each class**, to classify Z as the **most probable** class.

Logistic Regression

- It is not a Bayesian classifier, but it is a **probabilistic** one;
- Used to solve **binary classification** problems;
- Discover the **most probable class** for a new record;
- **Goal**: estimate the exponent z in order to maximize the **negative log-likelihood**, which is equivalent to **minimize the error**:

$$\hat{y} = \underset{c_i \in C}{\text{argmax}} P(c_i | x)$$

- **Gradient descent** is used to find the minimum of the error function.