# Bayesian Classification

Yufei Tao

Department of Computer Science and Engineering
Chinese University of Hong Kong

## Recall: Classification

Let $A_1, ..., A_d$ be the **attributes** of a $d$-dimensional universe $U$, i.e.:

$$U = dom(A_1) \times dom(A_2) \times ... \times dom(A_d)$$

where $dom(A_i)$ represents the set of possible values on $A_i$.

Each **object** is an element $o$ of $U \times \{0, 1\}$, i.e., it takes a value $o[A_i]$ on every attribute $A_i$ $(1 \leq i \leq d)$, and a **class label** $o[C]$ either 0 or 1.

Denote by $D$ a probabilistic distribution on $U \times \{0, 1\}$.

## Recall: Classification

**Goal:** Given an object $o$ drawn from $D$, we want to predict its label $o[C]$ from its attribute values $o[A_1], ..., o[A_d]$.

We do so by constructing a function

$$M : dom(A_1) \times dom(A_2) \times ... \times dom(A_d) \rightarrow \{0, 1\}$$

which we refer to as a **classifier**. Given any object $o$, we predict its class label as $M(o[A_1], ..., o[A_d])$.

We define the error of $M$ on $D$—denoted as $err_D(M)$—as:

$$err_D(M) \quad = \quad \boldsymbol{Pr}_{e \sim D}[M(o[A_1], ..., o[A_d]) \neq o[C]].$$

Namely, if we draw an object $o$ according to $D$, what is the probability that $M$ makes a mistake about the class label of $o$?

Ideally, we want to find an $M$ to minimize $err_D(M)$.

## Recall: Classification

In training, we are given a sample set $R$ of $D$, where each object in $R$ is drawn independently according to $D$. We refer to $R$ as the **training set**.

We would like to learn our classifier $M$ from $R$.

**Example:** Suppose that we have the following traning set:

| age | education | occupation | loan default |
|---|---|---|---|
| 28 | high school | self-employed | yes |
| 32 | master | programmer | no |
| 33 | undergrad | lawyer | yes |
| 37 | undergrad | programmer | no |
| 38 | undergrad | self-employed | yes |
| 45 | master | self-employed | no |
| 48 | high school | programmer | no |
| 50 | master | laywer | no |
| 52 | master | programmer | no |
| 55 | high school | self-employed | no |

Now we are given a new customer (31, high-school, lawyer) with an unknown "default" value. How should we predict this value?

This lecture will introduce another classification technique called Bayesian classification which, as the name suggests, finds its root in the Bayes' theorem.

The method works most effectively when each attribute has a small domain, namely, the attribute has only a small number of possible values. When an attribute has a large domain, we may reduce its domain size through discretization, which combines several values into one.

For example, in the training set of Slide 5, the attribute age has a large domain. We may discretize it into a smaller domain: $\{20^+, 30^+, 40^+, 50^+\}$, where "$20^+$" corresponds to the interval $[20, 29]$, "$30^+$" to $[30, 39]$, and so on. The next slide shows the resulting training set after the conversion.

**Example:** The training set after discretizing ages:

| age | education | occupation | loan default |
|---|---|---|---|
| $20^+$ | high school | self-employed | yes |
| $30^+$ | master | programmer | no |
| $30^+$ | undergrad | lawyer | yes |
| $30^+$ | undergrad | programmer | no |
| $30^+$ | undergrad | self-employed | yes |
| $40^+$ | master | self-employed | no |
| $40^+$ | high school | programmer | no |
| $50^+$ | master | laywer | no |
| $50^+$ | master | programmer | no |
| $50^+$ | high school | self-employed | no |

## Conditional probability:

Let $X$ and $Y$ be two events. Then, $\boldsymbol{Pr}[X \mid Y]$ is defined to be the probability of $X$ occurring, knowing that $Y$ has already occurred.

### Bayes' Theorem:

$$\boldsymbol{Pr}[X \mid Y] = \frac{\boldsymbol{Pr}[Y \mid X] \cdot \boldsymbol{Pr}[X]}{\boldsymbol{Pr}[Y]}$$

Now let us rephrase the task of classification from the perspective of conditional probabilities. Recall that, given an object $o$ drawn from the underlying distribution $D$, we aim to predict the class label $o[C]$ accurately from its attributes $o[A_1], ..., o[A_d]$.

The conditional probability $\boldsymbol{Pr}[o[C] = yes \mid o[A_1], ..., o[A_d]]$ can be interpreted intuitively as the percentage of the objects with class label "yes", among all those objects drawn from $D$ that share the same attribute values as $o$.

$\boldsymbol{Pr}[o[C] = no \mid o[A_1], ..., o[A_d]]$ can also be interpreted similarly with respect to the "no" label.

It is clear that we should classify $o$ as "yes" if

$$\boldsymbol{Pr}[o[C] = yes \mid o[A_1], ..., o[A_d]] \quad \geq \quad \boldsymbol{Pr}[o[C] = no \mid o[A_1], ..., o[A_d]]$$

and "no" otherwise.

Applying Bayes' theorem, we get:

$$\begin{aligned}&\boldsymbol{Pr}[o[C] = yes \mid o[A_1], ..., o[A_d]] \\ &= \frac{\boldsymbol{Pr}[o[A_1], ..., o[A_d] \mid o[C] = yes] \cdot \boldsymbol{Pr}[o[C] = yes]}{\boldsymbol{Pr}[o[A_1], ..., o[A_d]]}.\end{aligned}$$

Similarly, we get

$$\begin{aligned}&\boldsymbol{Pr}[o[C] = no \mid o[A_1], ..., o[A_d]] \\ &= \frac{\boldsymbol{Pr}[o[A_1], ..., o[A_d] \mid o[C] = no] \cdot \boldsymbol{Pr}[o[C] = no]}{\boldsymbol{Pr}[o[A_1], ..., o[A_d]]}.\end{aligned}$$

Thus, essentially the goal is to decide which one is larger:
$\boldsymbol{Pr}[o[A_1], ..., o[A_d] \mid o[C] = yes] \cdot \boldsymbol{Pr}[o[C] = yes]$, or
$\boldsymbol{Pr}[o[A_1], ..., o[A_d] \mid o[C] = no] \cdot \boldsymbol{Pr}[o[C] = no]$.

Bayesian classification estimates
$\boldsymbol{Pr}[o[A_1], ..., o[A_d] \mid o[C] = yes] \cdot \boldsymbol{Pr}[o[C] = yes]$ and
$\boldsymbol{Pr}[o[A_1], ..., o[A_d] \mid o[C] = no] \cdot \boldsymbol{Pr}[o[C] = no]$ using the training set.
Next, we will explain only the former, because the estimate of the latter is similar.

The objective, obviously, is to estimate two terms:

- $\boldsymbol{Pr}[o[C] = yes]$
- $\boldsymbol{Pr}[o[A_1], ..., o[A_d] \mid o[C] = yes]$

We will discuss each term in turn.

$\boldsymbol{Pr}[o[C] = yes]$

This is essentially the percentage of yes objects drawn from $D$.

Naturally, we estimate $\boldsymbol{Pr}[o[C] = yes]$ as the percentage of yes objects in the training set $R$.

**Example:** In Slide 7, $\boldsymbol{Pr}[o[C] = yes] = 0.3$.

$\boldsymbol{Pr}[o[A_1], ..., o[A_d] \mid o[C] = yes]$

This is essentially the percentage of objects with attribute values $o[A_1], ..., o[A_d]$ among all the yes objects in $R$.

If we carry on the rationale on the previous slide, we would estimate $\boldsymbol{Pr}[o[A_1], ..., o[A_d] \mid o[C] = yes]$ as the percentage of objects with attribute values $o[A_1], ..., o[A_d]$ among all the yes objects in $R$. But this is a bad idea because $R$ may have **very few** such objects, rendering the estimate unreliable (losing statistical significance).

This situation forces us to introduce assumptions which, if satisfied, will give us the statistical significance to evaluate probabilities like $\boldsymbol{Pr}[o[A_1], ..., o[A_d] \mid o[C] = yes]$.

## $\boldsymbol{Pr}[o[A_1], ..., o[A_d] \mid o[C] = yes]$ (cont.)

Bayesian classification makes an assumption here:

$$\boldsymbol{Pr}[o[A_1], ..., o[A_d] \mid o[C] = yes] \quad = \quad \prod_{i=1}^{d} \boldsymbol{Pr}[o[A_i] \mid o[C] = yes].$$

Then for each $i \in [1, d]$, we estimate $\boldsymbol{Pr}[o[A_i] \mid o[C] = yes]$ as the percentage of objects with attribute value $o[A_i]$ among all the yes objects in $R$.

**Example:** In Slide 7, $\boldsymbol{Pr}[30+, \text{high-school}, \text{lawyer} \mid o[C] = yes]$ is assumed to be the product of

- $\boldsymbol{Pr}[30+ \mid o[C] = yes]$, which is estimated as $2/3$
- $\boldsymbol{Pr}[\text{high-school} \mid o[C] = yes]$, which is estimated as $1/3$
- $\boldsymbol{Pr}[\text{lawyer} \mid o[C] = yes]$, which is estimated as $1/3$.

The product equals $2/27$.

## $\boldsymbol{Pr}[o[A_1], ..., o[A_d] \mid o[C] = yes]$ **(cont.)**

In estimating $\boldsymbol{Pr}[o[A_i] \mid o[C] = yes]$, it is possible that no yes object in $R$ has attribute value $o[A_i]$. This would force our estimate to be 0, which is undesired (think: why). Instead, we replace the 0 estimate with a very small value, for example, 0.000001.

**Example:** In Slide 7, $\boldsymbol{Pr}[\text{programmer} \mid o[C] = yes]$ is estimated as 0.000001.

**Think:** At the beginning, we said that Bayesian classification works better on small domains. Why?

It should be clear that the effectiveness of Bayesian classification relies on the accuracy of the assumption:

$$\boldsymbol{Pr}[o[A_1], ..., o[A_d] \mid o[C] = yes] \quad = \quad \prod_{i=1}^{d} \boldsymbol{Pr}[o[A_i] \mid o[C] = yes].$$

This assumption is called the conditional independence assumption (think: why?). When this assumption is seriously violated, the accuracy of the method drops significantly.

Recall that conditional independence was introduced to give us statistical significance in evaluating a probability. However, the form of conditional independence we used is too strong, and may not necessarily hold in practice. For this reason, the form of Bayes classifiers we have discussed is usually referred to as **naive Bayes classifiers**.

There are many other variants of Bayes classifiers which differ in the assumptions made in order to overcome statistical insignificance in evaluating a probability. For example, in the so-called **Bayes networks**, the main idea is to use a less stringent assumption of conditional independence as we show next.

Consider again the evaluation of
$\boldsymbol{Pr}[30+, \text{undergrad}, \text{programmer} \mid o[C] = no]$ in the context of Slide 7.

It would be reasonable to assume that Age and Education are independent from each other, given "Occupation = programmer" and "$C$ = no". Then, we can derive the probability as:

$$
\begin{aligned}
& \boldsymbol{Pr}[30+, \text{undergrad}, \text{programmer} \mid o[C] = no] \\
= \; & \boldsymbol{Pr}[30+, \text{undergrad} \mid \text{programmer}, o[C] = no] \cdot \\
& \boldsymbol{Pr}[\text{programmer} \mid o[C] = no] \\
= \; & \boldsymbol{Pr}[30+ \mid \text{programmer}, o[C] = no] \\
& \cdot \boldsymbol{Pr}[\text{undergrad} \mid \text{programmer}, o[C] = no] \\
& \cdot \boldsymbol{Pr}[\text{programmer} \mid o[C] = no] \\
= \; & \frac{2}{4} \cdot \frac{1}{4} \cdot \frac{4}{7} = 1/14.
\end{aligned}
$$

The conditional independence assumption from the previous slide can be summarized as:

$$\begin{aligned} & \boldsymbol{Pr}\Big[o[\mathrm{Age}], o[\mathrm{Edu}] \mid o[\mathrm{Occ}], o[C]\Big] \\ = \quad & \boldsymbol{Pr}\Big[o[\mathrm{Age}] \mid o[\mathrm{Occ}], o[C]\Big] \cdot \boldsymbol{Pr}\Big[o[\mathrm{Edu}] \mid o[\mathrm{Occ}], o[C]\Big] \end{aligned}$$

Assumptions like the above require domain knowledge from experts to formulate. In practice, there are typically multiple such rules applied on a training dataset.

In summary, Bayesian classification has a solid foundation on probability theory. However, its application in practice is often hampered by the lack of training data to evaluate the probability of a joint event that involves multiple (conjunctive) conditions. This issue can often be alleviated with appropriate conditional independence assumptions which are set by domain experts. The accuracy of classification depends heavily on how well those assumptions are satisfied.