Bayesian Classification

Yufei Tao

Department of Computer Science and Engineering
Chinese University of Hong Kong

Review: Classification

Let $A_1, ..., A_d$ be d **attributes**.

Instance space: $\mathcal{X} = dom(A_1) \times dom(A_2) \times ... \times dom(A_d)$ where $dom(A_i)$ represents the set of possible values on A_i.

Label space: $\mathcal{Y} = \{-1, 1\}$ (where -1 and 1 are **class labels**).

Instance-label pair (a.k.a. **object**): a pair $(\boldsymbol{x}, y)$ in $\mathcal{X} \times \mathcal{Y}$.

- $\boldsymbol{x}$ is a vector; we use $\boldsymbol{x}[A_i]$ to represent the vector's value on A_i $(1 \leq i \leq d)$.

Denote by $\mathcal{D}$ a probabilistic distribution over $\mathcal{X} \times \mathcal{Y}$.

Review: Classification

Goal: Given an object $(\boldsymbol{x}, y)$ drawn from $\mathcal{D}$, we want to predict its label y from its attribute values $\boldsymbol{x}[A_1], ..., \boldsymbol{x}[A_d]$.

Classifier (**hypothesis**): A function $h : \mathcal{X} \to \mathcal{Y}$.

Error of h **on** $\mathcal{D}$: $err_{\mathcal{D}}(h) = \boldsymbol{Pr}_{(\boldsymbol{x},y)\sim\mathcal{D}}[h(\boldsymbol{x}) \neq y]$.
namely, if we draw an object $(\boldsymbol{x}, y)$ according to $\mathcal{D}$, what is the probability that h mis-predicts the label?

We would like to learn a classifier h with small $err_{\mathcal{D}}(h)$ from a **training set** S where each object is drawn independently from $\mathcal{D}$.

The Ideal Classifier

Fix a point $\boldsymbol{p}$ in the instance space. **Think:** given a class label $c \in \mathcal{Y}$, how would you interpret the conditional probability

$$\boldsymbol{Pr}_{(\boldsymbol{x},y)\sim\mathcal{D}}[y = c \mid \boldsymbol{x} = \boldsymbol{p}]?$$

Design a classifier h_{opt} as follows:

- $h_{opt}(\boldsymbol{p}) = -1$ if $\boldsymbol{Pr}_{(\boldsymbol{x},y)\sim\mathcal{D}}[y = -1 \mid \boldsymbol{x} = \boldsymbol{p}] \geq 0.5$;
- $h_{opt}(\boldsymbol{p}) = 1$ otherwise.

This is the best classifier possible.

Its error on $\mathcal{D}$, namely, $err_{\mathcal{D}}(h_{opt})$, is the **bayesian error**.

We will introduce the **Bayesian method**, which aims to follow the decisions of h_{opt} by approximating the value of $\boldsymbol{Pr}_{(\boldsymbol{x},y)\sim\mathcal{D}}[y = c \mid \boldsymbol{x} = \boldsymbol{p}]$.

Henceforth, we will abbreviate $\boldsymbol{Pr}_{(\boldsymbol{x},y)\sim\mathcal{D}}[y = c \mid \boldsymbol{x} = \boldsymbol{p}]$ simply as $\boldsymbol{Pr}[y = c \mid \boldsymbol{p}]$.

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	programmer	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

Bayesian classification 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**.

For example, we may discretize the "age" attribute into a smaller domain: $\{20^+, 30^+, 40^+, 50^+\}$, where "$20^+$" corresponds to the interval $[20, 29]$, "30^+" to $[30, 39]$, and so on. See the next slide for the training set after the conversion.

Example: The training set after discretizing “age”:

age	education	occupation	loan default
20^+	high school	self-employed	yes
30^+	master	programmer	no
30^+	undergrad	programmer	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

Bayes' Theorem:

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

Given an instance $\boldsymbol{x}$, (as in h_{opt}) we predict its label as -1 if and only if

$$\boldsymbol{Pr}[y = -1 \mid \boldsymbol{x}] \quad \geq \quad \boldsymbol{Pr}[y = 1 \mid \boldsymbol{x}].$$

Applying Bayes' theorem, we get:

$$\boldsymbol{Pr}[y = 1 \mid \boldsymbol{x}] = \frac{\boldsymbol{Pr}[\mathbf{x} \mid y = 1] \cdot \boldsymbol{Pr}[y = 1]}{\boldsymbol{Pr}[\mathbf{x}]}.$$

Similarly:

$$\boldsymbol{Pr}[y = -1 \mid \boldsymbol{x}] = \frac{\boldsymbol{Pr}[\mathbf{x} \mid y = -1] \cdot \boldsymbol{Pr}[y = -1]}{\boldsymbol{Pr}[\mathbf{x}]}.$$

It suffices to decide which of the following is larger:

- $\boldsymbol{Pr}[\mathbf{x} \mid y = 1] \cdot \boldsymbol{Pr}[y = 1]$, or
- $\boldsymbol{Pr}[\mathbf{x} \mid y = -1] \cdot \boldsymbol{Pr}[y = -1]$.

Bayesian classification **estimates** $\boldsymbol{Pr}[\mathbf{x} \mid y = 1] \cdot \boldsymbol{Pr}[y = 1]$ and $\boldsymbol{Pr}[\mathbf{x} \mid y = -1] \cdot \boldsymbol{Pr}[y = -1]$ 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:

- $\boldsymbol{Pr}[y = 1]$
- $\boldsymbol{Pr}[\mathbf{x} \mid y = 1]$

We will discuss each term in turn.

$\boldsymbol{Pr}[y = 1]$

This is the probability for an object drawn from $\mathcal{D}$ to have label 1.

Naturally, we estimate $\boldsymbol{Pr}[y = 1]$ as the percentage of yes objects in the training set S.

Example: In Slide 8, $\boldsymbol{Pr}[y = 1] = 0.3$.

$\boldsymbol{Pr}[\mathbf{x} \mid y = 1]$

This is the probability for a "yes"-object drawn from $\mathcal{D}$ to carry exactly the attribute values $\mathbf{x}[A_1], ..., \mathbf{x}[A_d]$.

We could estimate $\boldsymbol{Pr}[\mathbf{x} \mid y = 1]$ as the percentage of objects having attribute values $\mathbf{x}[A_1], ..., \mathbf{x}[A_d]$ among all the yes objects in S. But this is a bad idea because S may have **very few (even none)** such objects, rendering the estimate unreliable (losing statistical significance).

This situation forces us to introduce assumptions which — if satisfied — would allow us to obtain a more reliable estimate of $\boldsymbol{Pr}[\mathbf{x} \mid y = 1]$.

$\boldsymbol{Pr}[\boldsymbol{x} \mid y = 1]$ (cont.)

Bayesian classification makes an **assumption** here:

$$\boldsymbol{Pr}[\boldsymbol{x} \mid y = 1] \quad = \quad \prod_{i=1}^{d} \boldsymbol{Pr}[\boldsymbol{x}[A_i] \mid y = 1].$$

For each $i \in [1, d]$, we estimate $\boldsymbol{Pr}[\boldsymbol{x}[A_i] \mid y = 1]$ as the percentage of objects with attribute value $\boldsymbol{x}[A_i]$ among all the yes objects in S.

Example: In Slide 8, $\boldsymbol{Pr}[30+, \text{high-school}, \text{programmer} \mid y = 1]$ is assumed to be the product of

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

The product equals $2/27$.

$\boldsymbol{Pr}[\mathbf{x} \mid y = 1]$ (cont.)

The estimate of $\boldsymbol{Pr}[\mathbf{x}[A_i] \mid y = 1]$ would be 0 if S does not have any yes-object with attribute value $\mathbf{x}[A_i]$. But that would force our estimate of $\boldsymbol{Pr}[\mathbf{x} \mid y = 1]$ to be 0. Instead, we replace the 0 estimate with a very small value, for example, 0.000001.

Example: In Slide 8, $\boldsymbol{Pr}[\text{lawyer} \mid y = 1]$ is estimated as 0.000001.

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

The effectiveness of Bayesian classification relies on the accuracy of the assumption:

$$\boldsymbol{Pr}[\mathbf{x} \mid y = 1] \;=\; \prod_{i=1}^{d} \boldsymbol{Pr}[\mathbf{x}[A_i] \mid y = 1].$$

This assumption is called the **conditional independence** assumption. When this assumption is seriously violated, the accuracy of the method drops significantly.

The approach we have discussed so far is known as **naive Bayes classification**.

The approach can be integrated with alternative (less stringent) conditional independence assumption. Consider the evaluation of

$$\boldsymbol{Pr}[30+, \text{undergrad}, \text{programmer} \mid y = -1]$$

in the context of Slide 8. Suppose that "age" and "education" are independent after fixing "occupation" and the class label. Then:

$$\begin{aligned}
& \boldsymbol{Pr}[30+, \text{undergrad}, \text{programmer} \mid y = -1] \\
= \quad & \boldsymbol{Pr}[30+, \text{undergrad} \mid \text{programmer}, y = -1] \cdot \\
& \boldsymbol{Pr}[\text{programmer} \mid y = -1] \\
= \quad & \boldsymbol{Pr}[30+ \mid \text{programmer}, y = -1] \\
& \cdot \boldsymbol{Pr}[\text{undergrad} \mid \text{programmer}, y = -1] \\
& \cdot \boldsymbol{Pr}[\text{programmer} \mid y = -1] \\
= \quad & \frac{2}{4} \cdot \frac{1}{4} \cdot \frac{4}{7} = 1/14.
\end{aligned}$$

Next, we will provide an alternative way to describe the Bayes method (using naive Bayes as an example). Our description will clarify what is actually the set $\mathcal{H}$ of classifiers to be learned from. This allows you to apply the generalization theorem (discussed in the previous lecture) to bound the generalization error of the classifier obtained.

Recall that we have attributes $A_1, ..., A_d$.
We assume that each A_i ($i \in [1, d]$) has a finite domain $dom(A_i)$.

For each A_i, we introduce $2|dom(A_i)|$ parameters. Specifically, for each value $a \in dom(A_i)$, there are two parameters:

- $p_i(a \mid -1)$, which is our estimate of $\boldsymbol{Pr}[\mathbf{x}[A_i] = a \mid y = -1]$;
- $p_i(a \mid 1)$, which is our estimate of $\boldsymbol{Pr}[\mathbf{x}[A_i] = a \mid y = 1]$.

Furthermore, we also introduce:

- $p(-1)$, which is our estimate of $\boldsymbol{Pr}[y = -1]$;
- $p(1)$, which is our estimate of $\boldsymbol{Pr}[y = 1]$.

In total, we have $2 + 2\sum_{i=1}^{d} |dom(A_i)|$ parameters.

Once the values of the $2 + 2\sum_{i=1}^{d} |dom(A_i)|$ parameters have been fixed, the conditional independence assumption (of naive Bayes) gives rise to the following classifier $h(\boldsymbol{x})$:

- $h(\boldsymbol{x}) = -1$ if

$$p(-1) \cdot \prod_{i=1}^{d} p_i(\boldsymbol{x}[A_i] \mid -1) \geq p(1) \cdot \prod_{i=1}^{d} p_i(\boldsymbol{x}[A_i] \mid 1)$$

- $h(\boldsymbol{x}) = 1$ otherwise.

The set $\mathcal{H}$ contains all such classifiers.

> **Remark:** The Bayes method we explained earlier gives an efficient way for choosing a reasonably good classifier $h \in \mathcal{H}$.

Next, we introduce the **Bayesian network** which is a popular way to describe sophisticated conditional independence assumptions.

Let us review some concepts on acyclic directed graphs (DAG):

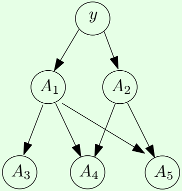
- A DAG G is a directed graph with no cycles.
- A node in G with 0 in-degree is a **root**. Note that G may have multiple roots.
- If a node u has an edge to another node v, then u is a **parent** of v. Note that a node can have multiple parents.

We will use $parents(v)$ to represent the set of parents of a node v.

We define a **Bayesian network** as an acyclic directed graph (DAG) G satisfying:

1. G has $d+1$ nodes, including a node for the class label and a node for each attribute;
2. G has a single root node, which must be the class label;
3. if attribute u has no path to attribute v, they are independent conditioned on $parents(u)$.

Example: The following is a Bayesian network with $d = 5$.



- A_1 and A_2 independent conditioned on y;
- A_4 and A_5 independent conditioned on A_1, A_2;
- A_3 and A_5 independent conditioned on A_1;
- A_3 and y independent conditioned on A_1.

Theorem 1: Given the conditional independence assumptions described by a Bayesian network G, we have

$$\boldsymbol{Pr}[A_1, ..., A_d \mid y] = \prod_{i=1}^{d} \boldsymbol{Pr}[A_i \mid parents(A_i)].$$

Before proving the theorem, let us first see an example.

Example: Given the Bayesian network on the previous slide, we have:

$$\boldsymbol{Pr}[A_1, A_2, ..., A_5 \mid y] =$$
$$\boldsymbol{Pr}[A_1 \mid y] \cdot \boldsymbol{Pr}[A_2 \mid y] \cdot \boldsymbol{Pr}[A_3 \mid A_1] \cdot \boldsymbol{Pr}[A_4 \mid A_1, A_2] \cdot \boldsymbol{Pr}[A_5 \mid A_1, A_2].$$

We will now proceed to prove the theorem. The following facts about conditional independence will be useful:

Lemma 1: If A and B are independent conditioned on C, then:

- $\boldsymbol{Pr}[A, B \mid C] = \boldsymbol{Pr}[A \mid C] \cdot \boldsymbol{Pr}[B \mid C]$;
- $\boldsymbol{Pr}[A \mid C, B] = \boldsymbol{Pr}[A \mid C]$.

Proof: The first bullet is the definition of conditional independence, whereas the second bullet holds because

$$\begin{aligned} \boldsymbol{Pr}[A \mid C, B] &= \frac{\boldsymbol{Pr}[A, B \mid C]}{\boldsymbol{Pr}[B \mid C]} \\ &= \frac{\boldsymbol{Pr}[A \mid C]\boldsymbol{Pr}[B \mid C]}{\boldsymbol{Pr}[B \mid C]} = \boldsymbol{Pr}[A \mid C]. \end{aligned}$$

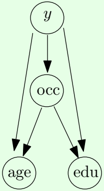
□

Proof of Theorem 1: Without loss of generality, suppose that y, A_1, ..., A_d is a topological order of G (namely, no path exists from a vertex u to any vertex before u).

$$\begin{aligned}
\boldsymbol{Pr}[A_1, ..., A_d \mid y] &= \boldsymbol{Pr}[A_2, ..., A_d \mid y, A_1] \cdot \boldsymbol{Pr}[A_1 \mid y] \\
&= \boldsymbol{Pr}[A_3, ..., A_d \mid y, A_1, A_2] \cdot \boldsymbol{Pr}[A_2 \mid y, A_1] \cdot \boldsymbol{Pr}[A_1 \mid y] \\
&\ldots \\
&= \prod_{i=1}^{d} \boldsymbol{Pr}[A_i \mid y, A_1, ..., A_{i-1}] \\
\text{(by Lemma 1)} &= \prod_{i=1}^{d} \boldsymbol{Pr}[A_i \mid parents(A_i)]
\end{aligned}$$

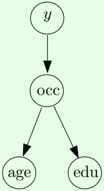
where the last equality used the conditional-independence properties implied by G and the fact that $parents(A_i) \subseteq \{y, A_1, ..., A_{i-1}\}$. □

Example: Consider the training set on Slide 8. If we are given the Bayesian network



then $\boldsymbol{Pr}[30+, \text{undergrad}, \text{programmer} \mid y = -1]$ is calculated as shown on Slide 17.

Example (cont.): If the Bayesian network is



then $\boldsymbol{Pr}[30+, \text{undergrad}, \text{programmer} \mid y = -1]$

$$\begin{aligned} &= \boldsymbol{Pr}[30+, \text{undergrad} \mid \text{programmer}, y = -1] \cdot \\ &\quad \boldsymbol{Pr}[\text{programmer} \mid y = -1] \\ &= \boldsymbol{Pr}[30+ \mid \text{programmer}] \cdot \boldsymbol{Pr}[\text{undergrad} \mid \text{programmer}] \\ &\quad \cdot \boldsymbol{Pr}[\text{programmer} \mid y = -1] \\ &= \frac{3}{5} \cdot \frac{2}{5} \cdot \frac{4}{7} = 24/175. \end{aligned}$$

Think: What is the set $\mathcal{H}$ of classifiers to be learned from if we are given the Bayesian network on the previous slide?