Probabilistic Classification

Based on the ML lecture by Raymond J. Mooney
University of Texas at Austin

Probabilistic Classification – Idea

Imagine that

- I look out of a window and see a bird,
- it is black, approx. 25 cm long, and has a rather yellow beak.

My daughter asks: What kind of bird is this?

My usual answer: This is *probably* a kind of blackbird (kos černý in Czech).

Here *probably* means that out of my extensive catalogue of four kinds of birds that I am able to recognize, "blackbird" gets the highest degree of belief based on *features* of this particular bird.

Frequentists might say that the largest proportion of birds with similar features I have ever seen were blackbirds.

The degree of belief (Bayesians), or the relative frequency (frequentists) is the *probability*.

Basic Discrete Probability Theory

- A finite or countably infinite set Ω of *possible outcomes*, Ω is called *sample space*.

  Experiment: Roll one dice once. Sample space: $\Omega = \{1, \ldots, 6\}$
- Each element ω of Ω is assigned a "probability" value $f(\omega)$, here f must satisfy
  - $f(\omega) \in [0, 1]$ for all $\omega \in \Omega$,
  - $\sum_{\omega \in \Omega} f(\omega) = 1$.

  If the dice is fair, then $f(\omega) = \frac{1}{6}$ for all $\omega \in \{1, \ldots, 6\}$.
- An *event* is any subset E of Ω.
- The *probability* of a given event $E \subseteq \Omega$ is defined as

$$P(E) = \sum_{\omega \in E} f(\omega)$$

  Let E be the event that an odd number is rolled, i.e., $E = \{1, 3, 5\}$. Then $P(E) = \frac{1}{2}$.
- Basic laws: $P(\Omega) = 1$, $P(\emptyset) = 0$, given disjoint sets A, B we have $P(A \cup B) = P(A) + P(B)$, $P(\Omega \setminus A) = 1 - P(A)$.

Conditional Probability and Independence

- $P(A \mid B)$ is the probability of A given B (assume $P(B) > 0$) defined by

  $$P(A \mid B) = P(A \cap B)/P(B)$$

  (We assume that B is all and only information known.)

  A fair dice: what is the probability that 3 is rolled assuming that an odd number is rolled? ... and assuming that an even number is rolled?

- **The law of total probability:** Let A be an event and $B_1, \ldots, B_n$ pairwise disjoint events such that $\Omega = \bigcup_{i=1}^{n} B_i$. Then

  $$P(A) = \sum_{i=1}^{n} P(A \cap B_i) = \sum_{i=1}^{n} P(A \mid B_i) \cdot P(B_i)$$

- A and B are independent if $P(A \cap B) = P(A) \cdot P(B)$.

  It is easy to show that if $P(B) > 0$, then
  A, B are independent iff $P(A \mid B) = P(A)$.

Random Variables

- A *random variable* X is a function $X : \Omega \to \mathbb{R}$.
  A dice: $X : \{1, \ldots, 6\} \to \{0, 1\}$ such that $X(n) = n \mod 2$.
- A *probability mass function (pmf)* of X is a function p defined by

$$p(x) := P(X = x)$$

Often $P(X)$ is used to denote the pmf of X.

Random Vectors

- A *random vector* is a function $X : \Omega \to \mathbb{R}^d$.

  We use $X = (X_1, \ldots, X_d)$ where X_i is a random variable returning the i-th component of X.
- A *joint probability mass function* of X is $p_X(x_1, \ldots, x_d) := P(X_1 = x_1 \wedge \cdots \wedge X_d = x_d)$.

  I.e., p_X gives the probability of every combination of values.

  Often, $P(X_1, \cdots, X_d)$ denotes the joint pmf of $X_1, \ldots, X_d$. That is, $P(X_1, \cdots, X_d)$ stands for probabilities $P(X_1 = x_1 \wedge \cdots \wedge X_d = x_d)$ for all possible combinations of $x_1, \ldots, x_d$.
- The probability mass function p_{X_i} of each X_i is called *marginal probability mass function*. We have

$$p_{X_i}(x_i) = P(X_i = x_i) = \sum_{(x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_d)} p_X(x_1, \ldots, x_d)$$

Random Vectors – Example

Let Ω be a space of colored geometric shapes that are divided into two categories (positive and negative).

Assume a random vector $X = (X_{color}, X_{shape}, X_{cat})$ where

- $X_{color} : \Omega \to \{red, blue\}$,
- $X_{shape} : \Omega \to \{circle, square\}$,
- $X_{cat} : \Omega \to \{pos, neg\}$.

The joint pmf is given by the following tables:

positive:

	circle	square
red	0.2	0.02
blue	0.02	0.01

negative:

	circle	square
red	0.05	0.3
blue	0.2	0.2

Random Vectors – Example

The probability of all possible events can be calculated by summing the appropriate probabilities.

$$\begin{aligned} P(red \wedge circle) &= P(X_{color} = red \ \wedge \ X_{shape} = circle) \\ &= P(red \wedge circle \wedge positive) + \\ &\quad + P(red \wedge circle \wedge negative) \\ &= 0.2 + 0.05 = 0.25 \end{aligned}$$

$$P(red) = 0.2 + 0.02 + 0.05 + 0.3 = 0.57$$

Thus also all conditional probabilities can be computed:

$$P(positive \mid red \wedge cicle) = \frac{P(positive \wedge red \wedge circle)}{P(red \wedge circle)} = \frac{0.2}{0.25} = 0.8$$

Conditional Probability Mass Functions

We often have to deal with a pmf of a random vector X conditioned on values of a random vector Y.

I.e., we are interested in $P(X = x \mid Y = y)$ for all x and y.

We write $P(X \mid Y)$ to denote the pmf of X conditioned on Y.

Technically, $P(X \mid Y)$ is a function which takes a possible value x of X and a possible value y of Y and returns $P(X = x \mid Y = y)$.

This allows us to say, e.g., that two variables X_1 and X_2 are independent conditioned on Y by

$$P(X_1, X_2 \mid Y) = P(X_1 \mid Y) \cdot P(X_2 \mid Y)$$

Technically this means that for all possible values x_1 of X_1, all possible values x_2 of X_2, and all possible values y of Y we have

$$P(X_1 = x_1 \wedge X_2 = x_2 \mid Y = y) = \\ P(X_1 = x_1 \mid Y = y) \cdot P(X_2 = x_2 \mid Y = y)$$

Bayesian Classification

Let Ω be a sample space (a universum) of all objects that can be classified.
We assume a probability P on Ω.

A *training set* will be a subset of Ω randomly sampled according to P.

- Let Y be the random variable for the category which takes values in $\{y_1, \ldots, y_m\}$.
- Let X be the random vector describing n features of a given instance, i.e., $X = (X_1, \ldots, X_n)$
  - Denote by x_k possible values of X,
  - and by x_{ij} possible values of X_i.

Bayes classifier: Given a vector of feature values x_k,

$$C^{Bayes}(x_k) := \arg\max_{i \in \{1,\ldots,m\}} P(Y = y_i \mid X = x_k)$$

Intuitively, C^{Bayes} assigns x_k to the most probable category it might be in.

Bayesian Classification – Example

Imagine a conveyor belt with apples and apricots.

A machine is supposed to correctly distinguish apples from apricots based on their weight and diameter.

That is,

- $Y = \{apple, apricot\}$,
- $X = (X_{weight}, X_{diam})$.

Assume that we are given a fruit that weighs $40g$ with $5cm$ diameter.

The Bayes classifier compares $P(Y = apple \mid X = (40g, 5cm))$ with $P(Y = apricot \mid X = (40g, 5cm))$ and selects the more probable category given the features.

Optimality of the Bayes Classifier

Let C be an arbitrary *classifier*, that is a function that to every x_k assigns a class out of $\{y_1, \ldots, y_m\}$.

Slightly abusing notation, we use C to also denote the random variable which assigns a category to every instance.

(Technically this is a composition $C \circ X$ of C and X which first determines the features using X and then classifies according to C).

Define the error of the classifier C by

$$E_C = P(Y \neq C)$$

Věta

The Bayes classifier C^{Bayes} minimizes E_C, that is

$$E_{C^{Bayes}} := \min_{C \text{ is a classifier}} E_C$$

Optimality of the Bayes Classifier

$$
\begin{aligned}
E_C &= \sum_{i=1}^{m} P(Y = y_i \wedge C \neq y_i) \\
&= 1 - \sum_{i=1}^{m} P(Y = y_i \wedge C = y_i) \\
&= 1 - \sum_{i=1}^{m} \sum_{x_k} P(Y = y_i \wedge C = y_i \mid X = x_k) P(X = x_k) \\
&= 1 - \sum_{x_k} P(X = x_k) \sum_{i=1}^{m} P(Y = y_i \wedge C = y_i \mid X = x_k) \\
&= 1 - \sum_{x_k} P(X = x_k) P(Y = C(x_k) \mid X = x_k)
\end{aligned}
$$

(Here the last equality follows from the fact that C is determined by x_k.)
Choosing

$$C(x_k) = C^{Bayes}(x_k) = \underset{i \in \{1,\ldots,m\}}{\arg\max}\; P(Y = y_i \mid X = x_k)$$

maximizes $P(Y = C(x_k) \mid X = x_k)$ and thus minimizes E_C.

Practical Use of Bayes Classifier

The crucial problem: How to compute $P(Y = y_i \mid X = x_k)$?

Given no other assumptions, this requires a table giving the probability of each category for each possible vector of feature values, which is impossible to accurately estimate from a reasonably-sized training set.

Concretely, if all $Y, X_1, \ldots, X_n$ are binary, we need 2^n numbers to specify $P(Y = 0 \mid X = x_k)$ for each possible x_k.
(Note that we do not need to specify
$P(Y = 1 \mid X = x_k) = 1 - P(Y = 0 \mid X = x_k)$).

It is a bit better than $2^{n+1} - 1$ entries for specification of the complete joint pmf $P(Y, X_1, \ldots, X_n)$.

However, it is still too large for most classification problems.

Let's Look at It the Other Way Round

Věta (Bayes,1764)

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

Důkaz.

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} = \frac{\frac{P(A \cap B)}{P(A)} \cdot P(A)}{P(B)} = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

$\square$

Bayesian Classification

Determine the category for x_k by finding y_i maximizing

$$P(Y = y_i \mid X = x_k) \quad = \quad \frac{P(Y = y_i) \cdot P(X = x_k \mid Y = y_i)}{P(X = x_k)}$$

So in order to make the classifier we need to compute:

- The prior $P(Y = y_i)$ for every y_i
- The conditionals $P(X = x_k \mid Y = y_i)$ for every x_k and y_i

Estimating the Prior and Conditionals

- $P(Y = y_i)$ can be easily estimated from data:
  - Given a set of p training examples where
  - n_i of the examples are in the category y_i,
  - we set

$$P(Y = y_i) = \frac{n_i}{p}$$

- If the dimension of features is small, $P(X = x_k \mid Y = y_i)$ can be estimated from data similarly as for $P(Y = y_i)$.

  Unfortunately, for higher dimensional data too many examples are needed to estimate all $P(X = x_k \mid Y = y_i)$ (there are too many x_k's).

  So where is the advantage of using the Bayes thm.?

We introduce *independence assumptions* about the features!

Generative Probabilistic Models

- Assume a simple (usually unrealistic) probabilistic method by which the data was generated.
- For classification, assume that each category y_i has a different parametrized generative model for $P(X = x_k \mid Y = y_i)$.
  - Maximum Likelihood Estiomation (MLE): Set parameters to maximize the probability that the model produced the given training data.
  - More conceretely: If M_λ denotes a model with parameter values λ, and D_k is the training data for the k-th category, find model parameters for category k (λ_k) that maximizes the likelihood of D_k :

$$\lambda_k = \arg\max_\lambda P(D_k \mid M_\lambda)$$

Generative Probabilistic Models – Simple Example

First, let us illustrate the generative model on a simple example.

Consider two binary features:

- $X_{color} : \Omega \to \{red, blue\}$
- $X_{shape} : \Omega \to \{circle, square\}$

and two classes $\{pos, neg\}$.

There are $2^3 = 8$ possible combinations of features and classes.

We assume that for each category, the features are distributed independently:

$$P(X_{color}, X_{shape} \mid pos) = P(X_{color} \mid pos) \cdot P(X_{shape} \mid pos)$$

$$P(X_{color}, X_{shape} \mid neg) = P(X_{color} \mid neg) \cdot P(X_{shape} \mid neg)$$

So we have to estimate four numbers (parameters):

$$P(red \mid pos), P(circle \mid pos), P(red \mid neg), P(circle \mid neg)$$

(As opposed to six when we want to completely specify the joint conditional pmfs.)

Generative Probabilistic Models – Simple Example

Given p training examples, assume that p_+ of them are positive, p_- of them are negative and that

- in ℓ^+_{red} *positive* examples the color is *red*,
- in ℓ^+_{circle} *positive* examples the shape is *circle*,
- in ℓ^-_{red} *negative* examples the color is *red*,
- in ℓ^-_{circle} *negative* examples the shape is *circle*.

Then MLE estimate $\bar{P}$ of P is

$$\bar{P}(red \mid pos) = \frac{\ell^+_{red}}{p_+} \qquad \bar{P}(circle \mid pos) = \frac{\ell^+_{circle}}{p_+}$$

$$\bar{P}(red \mid neg) = \frac{\ell^-_{red}}{p_-} \qquad \bar{P}(circle \mid neg) = \frac{\ell^-_{circle}}{p_-}$$

Now e.g. $\bar{P}(red \wedge circle \mid neg) = \frac{\ell^-_{red}}{p_-} \cdot \frac{\ell^-_{circle}}{p_-}$.

Note that if in reality the features are dependent, then the joint pmf **cannot** be obtained by such an estimate!

Naive Bayes

- We assume that features of an instance are (conditionally) independent *given the category*:

$$P(X \mid Y) = P(X_1, \cdots, X_n \mid Y) = \prod_{i=1}^{n} P(X_i \mid Y)$$

- Therefore, we only need to specify $P(X_i \mid Y)$, that is $P(X_i = x_{ij} \mid Y = y_k)$ for each possible pair of a feature-value x_{ij} and a class y_k.

  Note that if Y and all X_i are binary (values in $\{0, 1\}$), this requires specifying only $2n$ parameters:

$$P(X_i = 1 \mid Y = 1) \text{ and } P(X_i = 1 \mid Y = 0) \text{ for each } X_i$$

  since $P(X_i = 0 \mid Y) = 1 - P(X_i = 1 \mid Y)$.

Compared to specifying 2^n parameters without any independence assumptions.

Naive Bayes – Example

	positive	negative
P(Y)	0.5	0.5
$P(small \mid Y)$	0.4	0.4
$P(medium \mid Y)$	0.1	0.2
$P(large \mid Y)$	0.5	0.4
$P(red \mid Y)$	0.9	0.3
$P(blue \mid Y)$	0.05	0.3
$P(green \mid Y)$	0.05	0.4
$P(square \mid Y)$	0.05	0.4
$P(triangle \mid Y)$	0.05	0.3
$P(circle \mid Y)$	0.9	0.3

Is $(medium, red, circle)$ positive?

	positive	negative
P(Y)	0.5	0.5
$P(medium \mid Y)$	0.1	0.2
$P(red \mid Y)$	0.9	0.3
$P(circle \mid Y)$	0.9	0.3

Denote $x_k = (medium, red, circle)$.

$$\begin{aligned} P(pos \mid X = x_k) &= \\ &= P(pos) \cdot P(medium \mid pos) \cdot P(red \mid pos) \cdot P(circle \mid pos) \,/\, P(X = x_k) \\ &= 0.5 \cdot 0.1 \cdot 0.9 \cdot 0.9 \,/\, P(X = x_k) = 0.0405/P(X = x_k) \end{aligned}$$

$$\begin{aligned} P(neg \mid X = x_k) &= \\ &= P(neg) \cdot P(medium \mid neg) \cdot P(red \mid neg) \cdot P(circle \mid neg) \,/\, P(X = x_k) \\ &= 0.5 \cdot 0.2 \cdot 0.3 \cdot 0.3 \,/\, P(X = x_k) = 0.009/P(X = x_k) \end{aligned}$$

Apparently,

$$P(pos \mid X = x_k) = 0.0405/P(X = x_k) > 0.009/P(X = x_k) = P(neg \mid X = x_k)$$

So we classify x_k as positive.

Estimating Probabilities (In General)

- Normally, probabilities are estimated on observed frequencies in the training data (see the previous example).
- Let us have
  - n_k training examples in class y_k,
  - n_{ijk} of these n_k examples have the value for X_i equal to x_{ij}.

  Then we put $\bar{P}(X_i = x_{ij} \mid Y = y_k) = \frac{n_{ijk}}{n_k}$.
- **A problem:** If, by chance, a rare value x_{ij} of a feature X_i never occurs in the training data, we get

$$\bar{P}(X_i = x_{ij} \mid Y = y_k) = 0 \quad \text{for all } k \in \{1, \dots, m\}$$

  But then $\bar{P}(X = x_k) = 0$ for x_k containing the value x_{ij} for X_i, and thus $\bar{P}(Y = y_k \mid X = x_k)$ is not well defined. Moreover, $\bar{P}(Y = y_k) \cdot \bar{P}(X = x_k \mid Y = y_k) = 0$ (for all y_k) so even this cannot be used for classification.

Probability Estimation Example

Training data:

Size	Color	Shape	Class
small	red	circle	pos
large	red	circle	pos
small	red	triangle	neg
large	blue	circle	neg

Learned probabilities:

	positive	negative
$\bar{P}(Y)$	0.5	0.5
$\bar{P}(small \mid Y)$	0.5	0.5
$\bar{P}(medium \mid Y)$	0	0
$\bar{P}(large \mid Y)$	0.5	0.5
$\bar{P}(red \mid Y)$	1	0.5
$\bar{P}(blue \mid Y)$	0	0.5
$\bar{P}(green \mid Y)$	0	0
$\bar{P}(square \mid Y)$	0	0
$\bar{P}(triangle \mid Y)$	0	0.5
$\bar{P}(circle \mid Y)$	1	0.5

Note that $\bar{P}(medium \wedge red \wedge circle) = 0$.

So what is $\bar{P}(pos \mid medium \wedge red \wedge circle)$?

Smoothing

- To account for estimation from small samples, probability estimates are adjusted or *smoothed*.
- *Laplace smoothing* using an m-estimate works *as if*
  - each feature is given a prior probability p,
  - such feature have been observed with this probability p in a sample of size m (recall that m is the number of classes).

  We get

$$\bar{P}(X_i = x_{ij} \mid Y = y_k) = \frac{n_{ijk} + mp}{n_k + m}$$

(Recall that n_k is the number of training examples of class y_k, and n_{ijk} is the number of training examples of class y_k for which the i-th feature X_i has the value x_{ij}.)

Laplace Smothing Example

- Assume training set contains 10 positive examples:
  - 4 small
  - 0 medium
  - 6 large
- Estimate parameters as follows ($m = 2$ and $p = 1/3$)
  - $\bar{P}(small \mid positive) = (4 + 2/3)/(10 + 2) = 0.389$
  - $\bar{P}(medium \mid positive) = (0 + 2/3)/(10 + 2) = 0.056$
  - $\bar{P}(large \mid positive) = (6 + 2/3)/(10 + 2) = 0.556$

(We get
$\bar{P}(small \lor medium \lor large \mid positive) = 0.394 + 0.03 + 0.576 = 1.$)

Continuous Features

Ω may be (potentially) continuous, X_i may assign a continuum of values in $\mathbb{R}$.

- The probabilities are computed using *probability density* $p : \mathbb{R} \to \mathbb{R}^+$ instead of pmf.
  A random variable $X : \Omega \to \mathbb{R}^+$ has a density $p : \mathbb{R} \to \mathbb{R}^+$ if for every interval $[a, b]$ we have

  $$P(a \leq X \leq b) = \int_a^b p(x)dx$$

  Usually, $P(X_i \mid Y = y_k)$ is used to denote the *density* of X_i conditioned on $Y = y_k$.
- The densities $P(X_i \mid Y = y_k)$ are usually estimated using Gaussian densities as follows:
  - Estimate the mean μ_{ik} and the standard deviation σ_{ik} based on training data.
  - Then put

$$\bar{P}(X_i \mid Y = y_k) = \frac{1}{\sigma_{ik}\sqrt{2\pi}} \exp\left(\frac{-(X_i - \mu_{ik})^2}{2\sigma_{ik}^2}\right)$$

Comments on Naive Bayes

- Tends to work well despite rather strong assumption of conditional independence of features.
- Experiments show it to be quite competitive with other classification methods.

  Even if the probabilities are not accurately estimeted, it often picks the correct maximum probability category.
- Directly constructs a hypothesis from parameter estimates that are calculated from the training data.
- Consistency with the training data is not guaranteed.
- Typically handles noise well.
- Missing values are easy to deal with (simply average over all missing values in feature vectors).

Bayes Classifier vs MAP vs MLE

Recall that the Bayes classifier chooses the category as follows:

$$C^{Bayes}(x_k) = \underset{i \in \{1,\dots,m\}}{\arg\max}\; P(Y = y_i \mid X = x_k)$$
$$= \underset{i \in \{1,\dots,m\}}{\arg\max}\; \frac{P(Y = y_i) \cdot P(X = x_k \mid Y = y_i)}{P(X = x_k)}$$

As the denominator $P(X = x_k)$ is not influenced by i, the Bayes is equivalent to the Maximum Aposteriori Probability rule:

$$C^{MAP}(x_k) = \underset{i \in \{1,\dots,m\}}{\arg\max}\; P(Y = y_i) \cdot P(X = x_k \mid Y = y_i)$$

If we do not care about the prior (or assume uniform) we may use the Maximum Likelihood Estimate rule:

$$C^{MLE}(x_k) = \underset{i \in \{1,\dots,m\}}{\arg\max}\; P(X = x_k \mid Y = y_i)$$

(Intuitively, we maximize the probability that the data x_k have been generated into the category y_i.)

Bayesian Networks (Basic Information)

In the Naive Bayes we have assumed that *all* features $X_1, \ldots, X_n$ are independent.

This is usually not realistic.

E.g. Variables "rain" and "grass wet" are (usually) strongly dependent.

What if we return some dependencies back?

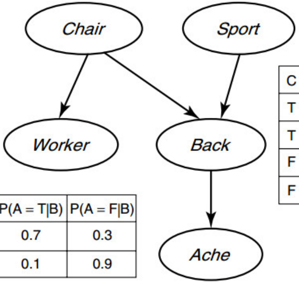
(But now in a well-defined sense.)

Bayesian networks are a graphical model that uses a directed acyclic graph to specify dependencies among variables.

Bayesian Networks – Example

P(C = T)	P(C = F)
0.8	0.2

P(S = T)	P(S = F)
0.02	0.98



C	P(W = T|C)	P(W = F|C)
T	0.9	0.1
F	0.01	0.99

C	S	P(B = T|C,S)	P(B = F|C,S)
T	T	0.9	0.1
T	F	0.2	0.8
F	T	0.9	0.1
F	F	0.01	0.99

B	P(A = T|B)	P(A = F|B)
T	0.7	0.3
F	0.1	0.9

Now, e.g.,

$$P(C, S, W, B, A) = P(C) \cdot P(S) \cdot P(W \mid C) \cdot P(B \mid C, S) \cdot P(A \mid B)$$

Now we may e.g. infer what is the probability $P(C = T \mid A = T)$ that we sit in a bad chair assuming that our back aches.

We have to store only 10 numbers as opposed to $2^5 - 1$ if the whole joint pmf is stored.

Bayesian Networks – Learning & Naive Bayes

Many algorithms have been developed for learning:

- the structure of the graph of the network,
- the *conditional probability tables*.

The methods are based on maximum-likelihood estimation, gradient descent, etc.

Automatic procedures are usually combined with expert knowledge.

Can you express the naive Bayes for $Y, X_1, \ldots, X_n$ using a Bayesian network?