Bayes' Theorem and Naive Bayes Classifier

Chapter · January 2018

DOI: 10.1016/B978-0-12-809633-8.20473-1

1 author:

Daniel Berrar
The Open University (UK)
88 PUBLICATIONS **2,068** CITATIONS

SEE PROFILE

Bayes' Theorem and Naive Bayes Classifier

Daniel Berrar

Data Science Laboratory
Tokyo Institute of Technology
2-12-1-S3-70 Ookayama, Meguro-ku, Tokyo 152-8550, Japan
Email: daniel.berrar@ict.e.titech.ac.jp

Abstract

The goal of this article is to give a mathematically rigorous yet easily accessible introduction to Bayes' theorem and the foundations of naive Bayes learning. Starting from fundamental elements of probability theory, this text outlines all steps leading to one of the oldest workhorses of machine learning: the *naive Bayes classifier*. As a tutorial, the text enables novice practitioners to quickly understand the essential concepts. As an encyclopedic article, the text provides a complete reference for bioinformaticians, machine learners, and statisticians, with an illustration of some caveats and pitfalls (and how to avoid them) in building a naive Bayes classifier in the R programming language.

Keywords: Alternative hypothesis, Bayes factor, Bayes' theorem, classification, likelihood, naive Bayes classifier, null hypothesis, total probability theorem

1. Introduction

Bayes' theorem is of fundamental importance for inferential statistics and many advanced machine learning models. Bayesian reasoning is a logical approach to updating the probability of hypotheses in the light of new evidence, and it therefore rightly plays a pivotal role in science [1]. Bayesian analysis allows us to answer questions for which frequentist statistical approaches were not developed. In fact, the very idea of assigning a probability to a hypothesis is not part of the frequentist paradigm.

The goal of this article is to provide both a mathematically rigorous yet concise explanation of the foundation of Bayesian statistics: *Bayes' theorem*, which underpins a simple but powerful machine learning algorithm: the *naive Bayes classifier* [2]. In contrast to other texts on these topics, this article is self-contained; it explains all terms and notations in detail and provides illustrative examples. As a tutorial, this text should therefore be easily accessible to readers from various backgrounds. As an encyclopedic article, it provides a complete reference for bioinformaticians, machine learners, and statisticians. Readers who are already familiar with the statistical background may find the practical examples in Section 3 most useful. Specifically, Section 3 highlights some caveats and pitfalls (and how to avoid them) in building a naive Bayes

This manuscript is the preprint of: Berrar D. (2018) Bayes' theorem and naive Bayes classifier. *Encyclopedia of Bioinformatics and Computational Biology*, Volume 1, Elsevier, pp. 403-412.

classifier using R, with additional materials available at the accompanying website `http://osf.io/92mes`.

2. Fundamentals

2.1. Basic Notation and Concepts

A statistical experiment can be broadly defined as a process that results in one and only one of several possible outcomes. The collection of all possible outcomes is called the *sample space*, denoted by Ω. At the introductory level, we can describe events by using notation from set theory. For example, our experiment may be one roll of a fair die. The sample space is then $\Omega = \{1, 2, 3, 4, 5, 6\}$, which is also referred to as *universal set* in the terminology of set theory. A simple event is, for instance, the outcome 2, which we denote as $E_1 = \{2\}$. The probability of an event is denoted by $P(\cdot)$. According to the classic concept of probability, the probability of an event E is the number of outcomes that are favorable to this event, divided by the total number of possible outcomes for the experiment, $P(E) = \frac{|E|}{|\Omega|}$, where $|E|$ denotes the cardinality of the set E, i.e., the number of elements in E. In our example, the probability of rolling a 2 is $P(E_1) = \frac{|E_1|}{|\Omega|} = \frac{1}{6}$. The event "the number is even" is a compound event, denoted by $E_2 = \{2, 4, 6\}$. The cardinality of E_2 is 3, so the probability of this event is $P(E_2) = \frac{3}{6}$.

The *complement* of E is the event that E does not occur and is denoted by E^c, with $P(E) = 1 - P(E^c)$. In the example, $E_2^c = \{1, 3, 5\}$.[1] Furthermore, $P(A|B)$ denotes the *conditional probability* of A given B. Finally, $\emptyset$ denotes the *empty set*, i.e., $\emptyset = \{\}$.

Let A and B be two events from a sample space Ω, which is either finite with N elements or countably infinite. Let $P : \Omega \rightarrow [0, 1]$ be a probability distribution on Ω, such that $0 < P(A) < 1$ and $0 < P(B) < 1$ and, obviously, $P(\Omega) = 1$. We can represent these events in a Venn diagram (Fig. 1a). The *union* of the events A and B, denoted by $A \cup B$, is the event that either A or B or both occur. The *intersection* of the events A and B, denoted by $A \cap B$, is the event that both A and B occur. Finally, two events, A and B, are called *mutually exclusive* if the occurrence of one of these events rules out the possibility of occurrence of the other event. In the notation of set theory, this means that A and B are disjoint, i.e., $A \cap B = \emptyset$. Two events A and B, with $P(A) > 0$ and $P(B) > 0$, are called *independent* if the occurrence of one event does not affect the probability of occurrence of the other event, i.e. $P(A|B) = P(A)$ or $P(B|A) = P(B)$, and $P(A \cap B) = P(A) \cdot P(B)$.

Note that the *conditional probability*, $P(A|B)$, is the *joint probability* $P(A \cap B)$ divided by the *marginal probability* $P(B)$. This is a fundamental relation, which has a simple geometrical interpretation. Loosely speaking, given that we are in the ellipse B (Fig. 1a), what is the probability that we are also in A? To be also in A, we have to be in the intersection $A \cap B$. Hence, the probability is equivalent to the number of elements in the intersection, $|A \cap B|$, divided by the number of elements in B, i.e., $|B|$. Formally, $P(A|B) = \frac{|A \cap B|}{|B|} = \frac{|A \cap B|/|\Omega|}{|B|/|\Omega|} = \frac{P(A \cap B)}{P(B)}$.

[1] In the literature, the complement of an event A is also often represented by the symbol $\bar{A}$.

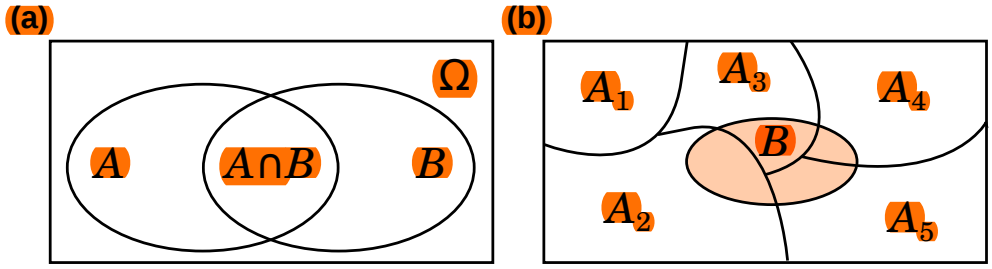

Figure 1: (a) Venn diagram for sets A and B. (b) Illustration of the total probability theorem. The sample space Ω is divided into five disjoint sets A_1 to A_5, which partly overlap with set B.

2.2. Total Probability Theorem

Before deriving Bayes' theorem, it is useful to consider the *total probability theorem*. First, the addition rule for two events, A and B, is easily derived from Figure 1a:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B) \tag{1}$$

We assume that the sample space can be divided into n mutually exclusive events A_i, $i = 1..n$, as shown in Figure 1(b). Specifically,

1. $A_1 \cup A_2 \cup \cdots \cup A_n = \Omega$
2. $A_i \cap A_j = \emptyset$ for $i \neq j$
3. $A_i \neq \emptyset$

From Figure 1(b), it is obvious that B can be stated as

$$B = (B \cap A_1) \cup (B \cap A_2) \cup \cdots \cup (B \cap A_n)$$

and we obtain the total probability theorem as

$$\begin{aligned} P(B) &= P(B \cap A_1) + P(B \cap A_2) + \cdots + P(B \cap A_n) - \underbrace{P(B \cap A_1 \cap \cdots \cap B \cap A_n)}_{=0, \text{ because } A_i \cap A_j = \emptyset \text{ for } i \neq j} \\ &= P(B|A_1)P(A_1) + P(B|A_2)P(A_2) + \cdots + P(B|A_n)P(A_n) \\ &= \sum_{i=1}^{n} P(B|A_i)P(A_i) \end{aligned} \tag{2}$$

which can be rewritten as

$$P(B) = P(B|A)P(A) + P(B|A^c)P(A^c) \tag{3}$$

because $A_2 \cup A_3 \cup ... \cup A_n$ is the complement of A_1 (cf. conditions 1 and 2 above). Redefining $A := A_1$ and $A^c := A_2 \cup A_3 \cup ... \cup A_n$ gives Eq. 3.

2.3. Bayes' Theorem

Assuming that $|A| \neq 0$ and $|B| \neq 0$, we can state the following:

$$P(A|B) = \frac{|A \cap B|}{|B|} = \frac{\frac{|A \cap B|}{|\Omega|}}{\frac{|B|}{|\Omega|}} = \frac{P(A \cap B)}{P(B)} \tag{4}$$

$$P(B|A) = \frac{|B \cap A|}{|A|} = \frac{\frac{|B \cap A|}{|\Omega|}}{\frac{|A|}{|\Omega|}} = \frac{P(A \cap B)}{P(A)} \tag{5}$$

From Eq. 4 and Eq. 5, it is immediately obvious that

$$P(A \cap B) = P(A|B)P(B) = P(B|A)P(A) \tag{6}$$

and therefore

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{7}$$

which is the simplest (and perhaps the most memorable) formulation of Bayes' theorem.

If the sample space Ω can be divided into finitely many mutually exclusive events $A_1, A_2, ...A_n$, and if B is an event with $P(B) > 0$, which is a subset of the union of all A_i, then for each A_i, the generalized Bayes' formula is

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum_{j=1}^{n} P(B|A_j)P(A_j)} \tag{8}$$

which can be rewritten as

$$P(A|B) = \frac{P(B|A)P(A)}{P(B|A)P(A) + P(B|A^c)P(A^c)} \tag{9}$$

Both Eq. 8 and Eq. 9 follow from Eq. 7 because of the total probability theorem (Eq. 2 and Eq. 3).

Bayes' theorem can be used to derive the posterior probability of a hypothesis given observed data:

$$P(\text{hypothesis}|\text{data}) = \frac{P(\text{data}|\text{hypothesis})P(\text{hypothesis})}{P(\text{data})} \tag{10}$$

where $P(\text{data}|\text{hypothesis})$ is the *likelihood* of the data given the hypothesis ("if the hypothesis is true, then what is the probability of observing these data?"), $P(\text{hypothesis})$ is the *prior probability* of the hypothesis ("what is the *a priori* probability of the hypothesis?"), and $P(\text{data})$ is the probability of observing the data, irrespective of the specified hypothesis. The prior probability (short, *prior*) is also referred to as the (initial) *degree of belief* in the hypothesis. In other words, the prior quantifies the *a priori* plausibility of the hypothesis.

It is often assumed that the data can arise under two competing hypotheses, H_1 and H_2, with $P(H_1) = 1 - P(H_2)$. Instead of "hypothesis", the term "model" is also frequently used. Let D denote the observed data. Then the posterior probability of the hypothesis (or model) H_1 is

$$P(H_1|D) = \frac{P(D|H_1)P(H_1)}{P(D|H_1)P(H_1) + P(D|H_2)P(H_2)} \tag{11}$$

and the posterior probability of H_2 is

$$P(H_2|D) = \frac{P(D|H_2)P(H_2)}{P(D|H_1)P(H_1) + P(D|H_2)P(H_2)} \tag{12}$$

From Eq. 11 and Eq. 12, we obtain

$$\underbrace{\frac{P(H_1|D)}{P(H_2|D)}}_{\text{posterior odds}} = \underbrace{\frac{P(D|H_1)}{P(D|H_2)}}_{\text{Bayes factor } B_{12}} \cdot \underbrace{\frac{P(H_1)}{P(H_2)}}_{\text{prior odds}} \tag{13}$$

The *Bayes factor* is the ratio of the posterior odds of H_1 to its prior odds. The Bayes factor can be interpreted as a summary measure of the evidence that the data provide us in favor of the hypothesis H_1 against its competing hypothesis H_2. If the prior probability of H_1 is the same as that of H_2 (i.e., $P(H_1) = P(H_2) = 0.5$), then the Bayes factor is the same as the posterior odds.

Note that in the simplest case, neither H_1 nor H_2 have any free parameters, and the Bayes factor then corresponds to the likelihood ratio [7]. If, however, at least one of the hypotheses (or models) has unknown parameters, then the conditional probabilities are obtained by integrating over the entire parameter space of H_i [7],

$$P(D|H_i) = \int P(D|\theta_i, H_i)P(\theta_i|H_i)d\theta_i \tag{14}$$

where θ_i denotes the parameters under H_i.

Note that Eq. 13 shows the Bayes factor B_{12} for only two hypotheses, but of course we may also consider more than just two. In that case, we can write B_{ij} to denote the Bayes factor for H_i against H_j. When only two hypotheses are considered, they are commonly referred to as *null hypothesis*, H_0, and *alternative hypothesis*, H_1. Jeffreys suggests grouping the values of B_{01} into grades [8] (Table 1):

It is instructive to compare the Bayes factor with the p-value from Fisherian significance testing. In short, the p-value is defined as the probability of obtaining a result as extreme as (or more extreme than) the actually observed result, given that the null hypothesis is true. The p-value is generally considered an evidential weight against the null hypothesis: the smaller the p-value, the greater the weight against H_0. However, the p-value can be a highly misleading measure of evidence because it overstates the evidence against H_0 [3, 4, 5]. A Bayesian calibration of p-values is described in [6]. This calibration leads to the

Table 1: Interpretation of Bayes factor B_{01} according to [8].

Grade	B_{01}	Interpretation
0	$B_{01} > 1$	Null hypothesis H_0 supported
1	$1 > B_{01} > 0.32$	Evidence against H_0, but not worth more than a bare mention
2	$0.32 > B_{01} > 0.10$	Evidence against H_0 substantial
3	$0.10 > B_{01} > 0.032$	Evidence against H_0 strong
4	$0.032 > B_{01} > 0.01$	Evidence against H_0 very strong
5	$0.01 > B_{01}$	Evidence against H_0 decisive

Bayes factor bound, $\bar{B} = \frac{1}{-ep \log p}$, where p is the p-value. Note that $\bar{B}$ is an upper bound on the Bayes factor over any reasonable choice of the prior distribution of the hypothesis "H_0 is not true", which we may refer to as "alternative hypothesis".[2] For example, a p-value of 0.01 corresponds to an odds of, at most, about 8 to 1 in favor of "H_0 is not true".

So far, we have considered only the discrete case, i.e., when the sample space is countable. What if the variables are continuous? Let X and Y denote two continuous random variables with joint probability density function $f_{XY}(x, y)$. Let $f_{X|Y}(x|y)$ and $f_{Y|X}(y|x)$ denote their conditional probability density functions. Then

$$f_{X|Y}(x|y) = \frac{f_{XY}(x, y)}{f_Y(y)} \tag{15}$$

and

$$f_{Y|X}(y|x) = \frac{f_{XY}(x, y)}{f_X(x)} \tag{16}$$

so that Bayes' theorem for continuous variables can be stated as

$$f_{X|Y}(x|y) = \frac{f_{Y|X}(y|x) f_X(x)}{f_Y(y)}, \tag{17}$$

where $f_Y(y) = \int_X f_{Y|X}(y|x) f_X(x) dx = \int_{-\infty}^{+\infty} f_{XY}(x, y) dx$ because of the total probability theorem.[3]

In summary, Bayes' theorem provides a logical method that combines new evidence (i.e., new data, new observations) with prior probabilities of hypotheses in order to obtain posterior probabilities for these hypotheses.

[2]Note that the concept of "alternative hypothesis" does not exist in the Fisherian significance testing, which considers only one hypothesis, i.e., the null hypothesis H_0. The idea of "alternative hypothesis" is firmly embedded in the Neyman-Pearsonian hypothesis testing, where the concept of the p-value does not exist. The two different schools of thought—the Fisherian and the Neyman-Pearsonian—should not be conflated; compare [4].

[3]Compare this with the discrete case: $P_X(x) = P_{X|Y}(x|y_1)P_Y(y_1) + \cdots + P_{X|Y}(x|y_n)P_Y(y_n) = \sum_y P_{X|Y}(x|y)P_Y(y) = \sum_y P_{XY}(x, y)$.

2.4. Naive Bayes Classifier

We assume that a data set contains n instances (or cases) $\mathbf{x}_i$, $i = 1..n$, which consist of p attributes, i.e., $\mathbf{x}_i = (x_{i1}, x_{i2}, ..., x_{ip})$. Each instance is assumed to belong to one (and only one) class $y \in \{y_1, y_2, ..., y_c\}$. Most predictive models in machine learning generate a numeric score s for each instance $\mathbf{x}_i$. This score quantifies the degree of class membership of that case in class y_j. If the data set contains only positive and negative instances, $y \in \{0, 1\}$, then a predictive model can either be used as a *ranker* or as a *classifier*. The ranker uses the scores to order the instances from the most to the least likely to be positive. By setting a threshold t on the ranking score, $s(\mathbf{x})$, such that $\{s(\mathbf{x}) \geq t\} = 1$, the ranker becomes a (crisp) classifier [9].

Naive Bayes learning refers to the construction of a Bayesian probabilistic model that assigns a posterior class probability to an instance: $P(Y = y_j|X = \mathbf{x}_i)$. The simple naive Bayes classifier uses these probabilities to assign an instance to a class. Applying Bayes' theorem (Eq. 7), and simplifying the notation a little, we obtain

$$P(y_j|\mathbf{x}_i) = \frac{P(\mathbf{x}_i|y_j)P(y_j)}{P(\mathbf{x}_i)} \tag{18}$$

Note that the numerator in Eq. 18 is the joint probability of $\mathbf{x}_i$ and y_j (cf. Eq. 6). The numerator can therefore be rewritten as follows; here, we will just use $\mathbf{x}$, omitting the index i for simplicity:

$$\begin{aligned}
P(\mathbf{x}|y_j)P(y_j) &= P(\mathbf{x}, y_j) \\
&= P(x_1, x_2, ..., x_p, y_j) \\
&= P(x_1|x_2, x_3, ..., x_p, y_j)P(x_2, x_3, ..., x_p, y_j) \quad \text{because} \quad P(a, b) = P(a|b)P(b) \\
&= P(x_1|x_2, x_3, ..., x_p, y_j)P(x_2|x_3, x_4, ..., x_p, y_j)P(x_3, x_4, ..., x_p, y_j) \\
&= P(x_1|x_2, x_3, ..., x_p, y_j)P(x_2|x_3, x_4, ..., x_p, y_j) \cdots P(x_p|y_j)P(y_j)
\end{aligned}$$

Let us assume that the individual x_i are independent from each other. This is a strong assumption, which is clearly violated in most practical applications and is therefore *naive*—hence the name. This assumption implies that $P(x_1|x_2, x_3, ..., x_p, y_j) = P(x_1|y_j)$, for example. Thus, the joint probability of $\mathbf{x}$ and y_j is

$$\begin{aligned}
P(\mathbf{x}|y_j)P(y_j) &= P(x_1|y_j) \cdot P(x_2|y_j) \cdots P(x_p|y_j)P(y_j) \\
&= \prod_{k=1}^{p} P(x_k|y_j)P(y_j)
\end{aligned} \tag{19}$$

which we can plug into Eq. 18 and we obtain

$$P(y_j|\mathbf{x}) = \frac{\prod_{k=1}^{p} P(x_k|y_j)P(y_j)}{P(\mathbf{x})} \tag{20}$$

Note that the denominator, $P(\mathbf{x})$, does not depend on the class—for example, it is the same for class y_j and y_l. $P(\mathbf{x})$ acts as a scaling factor and ensures that the posterior probability $P(y_j|\mathbf{x})$ is properly scaled

(i.e., a number between 0 and 1). When we are interested in a crisp classification rule, that is, a rule that assigns each instance to exactly one class, then we can simply calculate the value of the numerator for each class and select that class for which this value is maximal. This rule is called the *maximum posterior rule* (Eq. 21). The resulting "winning" class is also known as the *maximum a posteriori* (MAP) class, and it is calculated as $\hat{y}$ for the instance $\mathbf{x}$ as follows:

$$\hat{y} = \underset{y_j}{\operatorname{argmax}} \prod_{k=1}^{p} P(x_k|y_j)P(y_j) \tag{21}$$

A model that implements Eq. 21 is called a (*simple*) *naive Bayes classifier*.

A crisp classification, however, is often not desirable. For example, in ranking tasks involving a positive and a negative class, we are often more interested in how well a model ranks the cases of one class in relation to the cases of the other class [10]. The estimated class posterior probabilities are natural ranking scores. Applying again the total probability theorem (Eq. 3), we can rewrite Eq. 20 as

$$P(y_j|\mathbf{x}) = \frac{\prod_{k=1}^{p} P(x_k|y_j)P(y_j)}{\prod_{k=1}^{p} P(x_k|y_j)P(y_j) + \prod_{k=1}^{p} P(x_k|y_j^c)P(y_j^c)}. \tag{22}$$

3. Examples

3.1. Application of Bayes' Theorem in Medical Screening

Consider a population of people in which 1% really have a disease, D. A medical screening test is applied to 1000 randomly selected persons from that population. It is known that the sensitivity of the test is 0.90, and the specificity of the test is 0.91.

(a) If a tested person is really sick, then what is the probability of a positive test result (i.e., the result of the test indicates that the person is sick)?

(b) If the test is positive, then what is the probability that the person is really sick?

The probability that a randomly selected person has the disease is given as $P(D) = 0.01$ and thus $P(D^c) = 0.99$. These are the marginal probabilities that are known *a priori*, that is, without any knowledge of the person's test result. The sensitivity of a test is defined as $\frac{TP}{TP+FN}$, where TP denotes the number of true positive predictions and FN denotes the number of false negative predictions. Sensitivity is therefore also known as *true positive rate*; in information retrieval and data mining, it is also called *recall*. The specificity of a test is defined as $\frac{TN}{TN+FP}$, where TN denotes the number of true negative predictions and FP denotes the number of false positive predictions. Let $\oplus$ denote a positive and $\ominus$ a negative test result, respectively. The answer to (a) is therefore simple—in fact, it is already given: the conditional probability $P(\oplus|D)$ is the same as the sensitivity, since the number of persons who are really sick is the same as the number of true positive predictions (persons are sick and they correctly identified as such by the test) plus the number of false negative predictions (persons are sick but they are not identified as such by the test). Thus, $P(\oplus|D) = \frac{TP}{TP+FN} = 0.9$.

To answer (b), we use Bayes' theorem and obtain

$$P(D|\oplus) = \frac{P(\oplus|D)P(D)}{P(\oplus|D)P(D) + P(\oplus|D^c)P(D^c)} = \frac{0.9 \cdot 0.01}{0.9 \cdot 0.01 + 0.09 \cdot 0.99} = 0.092. \quad (23)$$

The only unknown in Eq. 23 is $P(\oplus|D^c)$, which we can easily derive from the given information: if the specificity is 0.91 or 91%, then the false positive rate must be 0.09 or 9%. But the false positive rate is the same as the conditional probability of a positive result, given the absence of disease, i.e., $P(\oplus|D^c) = 0.09$.

It can be insightful to represent the given information in a confusion matrix (Table 2). Here, the number of true negatives and false positives are rounded to the nearest integer. From the table, we can readily infer the chance of disease given a positive test result as $\frac{9}{9+89}$, i.e., just a bit more than 9%.

Table 2: Confusion table for the example on medical screening.

	D	D^c	Σ
$\oplus$	$TP = 9$	$FP = 89$	98
$\ominus$	$FN = 1$	$TN = 901$	902
Σ	10	990	1000

The conditional probability $P(D|\oplus)$ is also known as *positive predictive value* in epidemiology or as *precision* in data mining and related fields. What is the implication of this probability being around 0.09? The numbers in this example refer to health statistics for breast cancer screening with mammography [11]. A positive predictive value of just over 9% means that only about 1 out of every 10 women with a positive mammogram actually has breast cancer; the remaining 9 persons are falsely alarmed. Gigerenzer et al. showed that many gynecologists do not know the probability that a person has a disease given a positive test result, even when they are given appropriate health statistics framed as conditional probabilities [11]. By contrast, if the information is reframed in terms of natural frequencies (as in $\frac{9}{9+89}$ in this example), then the information is often easier to understand.

3.2. Naive Bayes Classifier – Introductory Example

We illustrate naive Bayes learning using the contrived data set[4] shown in Table 3. The first 14 instances refer to biological samples that belong to either the class *tumor* or the class *normal*. These samples represent the training set. Each instance is described by an expression profile of only four genes. Here, the gene expression values are discretized into either underexpressed (-1), overexpressed ($+1$), or normally expressed (0). Sample #15 represents a new biological sample. What is the likely class of this sample? Note that the particular combination of features, $\mathbf{x}_{15} = (+1, -1, +1, +1)$, does not appear in the training set.

Table 3: Contrived gene expression data set of 15 biological samples, each described by the discrete expression level of 4 genes. A sample belongs either to class "normal" or "tumor". Instance #15 is a new, unclassified sample.

Sample	Gene A	Gene B	Gene C	Gene D	Class
1	+1	+1	+1	0	normal
2	+1	+1	+1	+1	normal
3	0	+1	+1	0	tumor
4	−1	0	+1	0	tumor
5	−1	−1	0	0	tumor
6	−1	−1	0	+1	normal
7	0	−1	0	+1	tumor
8	+1	0	+1	0	normal
9	+1	−1	0	0	tumor
10	−1	0	0	0	tumor
11	+1	0	0	+1	tumor
12	0	0	+1	+1	tumor
13	0	+1	0	0	tumor
14	−1	0	+1	+1	normal
15	+1	−1	+1	+1	unknown

Using Eq. 20, we obtain

$$P(\text{tumor}|\mathbf{x}_{15}) = \frac{P(\text{A} = +1|\text{tumor}) \cdot P(\text{B} = -1|\text{tumor}) \cdot P(\text{C} = +1|\text{tumor}) \cdot P(\text{D} = +1|\text{tumor}) \cdot P(\text{tumor})}{P(\mathbf{x}_{15})}.$$

Let's begin with the prior probability of "tumor", $P(\text{tumor})$. This probability can be estimated as the fraction of tumor samples in the data set, i.e., $P(\text{tumor}) = \frac{9}{14}$. What is the fraction of samples for which gene A is overexpressed (+1), given that the class is "tumor"? As as an estimate for this conditional probability, $P(\text{Gene A} = +1|\text{tumor})$, the empirical value of $\frac{2}{9}$ (cf. samples #9 and #11) will be used.

Next, to calculate $P(\text{B} = -1|\text{tumor})$, we proceed as follows: among the nine tumor samples, for how many do we observe B = −1? We observe B = −1 for cases #5, #7, and #9, so the conditional probability is estimated as $\frac{3}{9}$. The remaining conditional probabilities are derived analogously. Thus, we obtain

$$P(\text{tumor}|\mathbf{x}_{15}) = \frac{\frac{2}{9} \cdot \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{9}{14}}{P(\mathbf{x}_{15})} = \frac{0.00529}{P(\mathbf{x}_{15})}$$

[4]This example is inspired by the famous Play Tennis data set, which is often used to illustrate naive Bayes learning in introductory data mining textbooks [12].

$$P(\text{normal}|\mathbf{x}_{15}) = \frac{\frac{3}{5} \cdot \frac{1}{5} \cdot \frac{4}{5} \cdot \frac{3}{5} \cdot \frac{5}{14}}{P(\mathbf{x}_{15})} = \frac{0.02057}{P(\mathbf{x}_{15})}.$$

With the denominator $P(\mathbf{x}_{15}) = 0.00529 + 0.02057$, we then obtain the properly scaled probabilities $P(\text{tumor}|\mathbf{x}_{15}) = 0.2046$ and $P(\text{normal}|\mathbf{x}_{15}) = 0.7954$.

3.3. Laplace Smoothing

When the number of samples is small, a problem may arise over how to correctly estimate the probability of an attribute given the class. Let us assume that at least one attribute value of the test instance, $\mathbf{x}$, is absent in all training instances of a class y_i. For example, assume that Gene A of instance #9 and #11 in Table 3 are underexpressed (-1) instead of overexpressed ($+1$). Then we obtain the following conditional probabilities,

$$\begin{aligned} P(\text{Gene A} = +1|\text{tumor}) &= \frac{0}{9} \\ P(\text{Gene A} = 0|\text{tumor}) &= \frac{4}{9} \\ P(\text{Gene A} = -1|\text{tumor}) &= \frac{5}{9} \end{aligned}$$

which obviously leads to $P(\text{tumor}|\mathbf{x}_{15}) = 0$. If Gene A is underexpressed (-1) in instances #9 and #11 in Table 3, then $P(\text{Gene A} = +1|\text{tumor}) = 0$, which implies that it is impossible to observe an overexpressed Gene A in a sample of class "tumor". Is it wise to make such a strong assumption? Probably not. It might be better to allow for a small, non-zero probability. This is what *Laplace smoothing* does [12]. In this example, we simply add 1 to each of the three numerators above and then add 3 to each of the denominators:

$$\begin{aligned} P(\text{Gene A} = +1|\text{tumor}) &= \frac{0+1}{9+3} \\ P(\text{Gene A} = 0|\text{tumor}) &= \frac{4+1}{9+3} \\ P(\text{Gene A} = -1|\text{tumor}) &= \frac{5+1}{9+3} \end{aligned}$$

However, instead of adding 1, we could also add a small positive constant c weighted by p_i,

$$\begin{aligned} P(\text{Gene A} = +1|\text{tumor}) &= \frac{0+cp_1}{9+c} \\ P(\text{Gene A} = 0|\text{tumor}) &= \frac{4+cp_2}{9+c} \\ P(\text{Gene A} = -1|\text{tumor}) &= \frac{5+cp_3}{9+c} \end{aligned}$$

with $p_1 + p_2 + p_3 = 1$, which are the prior probabilities for the states of expression for Gene A. Although such a fully Bayesian specification is possible, in practice, it is often unclear how the priors should be estimated, and simple Laplace is often appropriate [12].

3.4. Mixed Variables

In contrast to many other machine learning models, the naive Bayes classifier can easily cope with mixed-variable data sets. For example, consider Table 4. Here, Gene B has numeric expression values.

Table 4: Contrived gene expression data set from Table 3. Here, absolute expression values are reported for Gene B.

Sample	Gene A	Gene B	Gene C	Gene D	Class
1	+1	35	+1	0	normal
2	+1	30	+1	+1	normal
3	0	32	+1	0	tumor
4	−1	20	+1	0	tumor
5	−1	15	0	0	tumor
6	−1	13	0	+1	normal
7	0	11	0	+1	tumor
8	+1	22	+1	0	normal
9	+1	14	0	0	tumor
10	−1	24	0	0	tumor
11	+1	23	0	+1	tumor
12	0	25	+1	+1	tumor
13	0	33	0	0	tumor
14	−1	21	+1	+1	normal
15	+1	12	+1	+1	unknown

Assuming that the expression values of Gene B follow a normal distribution, we can model the probability density for class y_i as

$$f(x|y_i) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{1}{2}(\frac{x-\mu_i}{\sigma_i})^2} \tag{24}$$

where μ_i and σ_i denote the mean and standard deviation of the gene expression value for class y_i, respectively. Of course, in practice, other distributions are possible, and we need to choose the density that best describes the data. In the example, we obtain $\mu_{\text{tumor}} = 21.9$, $\sigma_{\text{tumor}} = 7.7$, and $\mu_{\text{normal}} = 24.2$, $\sigma_{\text{normal}} = 8.5$. Note that the probability that a continuous random variable X takes on a particular value is always zero for any continuous probability distribution, i.e., $P(X = x) = 0$. However, using the probability density function, we can calculate the probability that X lies in a narrow interval $[x_0 - \frac{\epsilon}{2}, x_0 + \frac{\epsilon}{2}]$ around x_0 as $\epsilon \cdot f(X = x_0)$. For the new instance $\mathbf{x}_{15}$ (Table 4), we obtain $f(12|\text{tumor}) = 0.02267$ and $f(12|\text{normal}) = 0.01676$, so that we can state the conditional probabilities as

$$P(\text{tumor}|\mathbf{x}_{15}) = \frac{\frac{2}{9} \cdot 0.0227\epsilon \cdot \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{9}{14}}{P(\mathbf{x}_{15})} = \frac{0.00036\epsilon}{P(\mathbf{x}_{15})} \quad \text{and}$$

$$P(\text{normal}|\mathbf{x}_{15}) = \frac{\frac{3}{5} \cdot 0.01676\epsilon \cdot \frac{4}{5} \cdot \frac{3}{5} \cdot \frac{5}{14}}{P(\mathbf{x}_{15})} = \frac{0.00172\epsilon}{P(\mathbf{x}_{15})}.$$

The posterior probabilities are $P(\text{tumor}|\mathbf{x}_{15}) = \frac{0.00036\epsilon}{0.00036\epsilon+0.00172\epsilon} = 0.17$ and $P(\text{normal}|\mathbf{x}_{15}) = \frac{0.00172\epsilon}{0.00036\epsilon+0.00172\epsilon} = 0.83$. Note that ϵ cancels.

3.5. Missing Value Imputation

Missing values do not present any problem for the naive Bayes classifier. Let us assume that the new instance contains missing values (encoded as NA), for example, $\mathbf{x}_{15} = (+1, \text{NA}, +1, +1)$. The posterior probability for class y_i can then be calculated by simply omitting this attribute, i.e.,

$$P(\text{tumor}|\mathbf{x}_{15}) = \frac{\frac{2}{9} \cdot \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{9}{14}}{P(\mathbf{x}_{15})} = \frac{0.016}{P(\mathbf{x}_{15})} \quad \text{and}$$

$$P(\text{normal}|\mathbf{x}_{15}) = \frac{\frac{3}{5} \cdot \frac{4}{5} \cdot \frac{3}{5} \cdot \frac{5}{14}}{P(\mathbf{x}_{15})} = \frac{0.103}{P(\mathbf{x}_{15})}$$

If the training set has missing values, then the conditional probabilities can be calculated by omitting these values. For example, suppose that the value +1 is missing for Gene A in sample #1 (Table 3). What is the probability that Gene A is overexpressed (+1), given that the sample is normal? There are five normal samples, and two of them (#2 and #8, Table 4) have an overexpressed Gene A. Therefore, the conditional probability is calculated as $P(\text{Gene A} = +1|\text{normal}) = \frac{2}{5}$.

3.6. R Implementation

We will now illustrate how to build a naive Bayes classifier using the function `naiveBayes()` of the package `e1073` [13] in the programming language and environment R [14], which is widely used by the bioinformatics community. Here, we use the data from Table 3.

```
library(e1071)
#' Load the data.
train <- read.csv("NB_train.csv", colClasses = c('factor', 'factor', 'factor', 'factor'))
test <- read.csv("NB_test.csv", colClasses = c('factor', 'factor', 'factor', 'factor'))
# Build the model.
NB_model <- naiveBayes(Class ~ ., data = train)
# Predict the test case.
```

```
pred <- predict(NB_model, test, type = "raw")
pred

##           normal     tumor
## [1,] 0.000202459 0.9997975
```

Surprisingly, these probabilities differ from what we calculated above, namely $P(\text{tumor}|\mathbf{x}_{15}) = 0.2046$ and $P(\text{normal}|\mathbf{x}_{15}) = 0.7954$. Why? The problem can be quite hard to spot. The reason is that the factor levels (per attribute) are not the same in the training and the test set, which causes `predict()` to calculate incorrect probabilities. Internally, `predict()` converts attribute values into numbers, and it does not check whether the factor levels are consistent or not.

```
str(train)

## 'data.frame': 14 obs. of  5 variables:
##  $ Gene_A: Factor w/ 3 levels "0","-1","+1": 3 3 1 2 2 2 1 3 3 2 ...
##  $ Gene_B: Factor w/ 3 levels "0","-1","+1": 3 3 3 1 2 2 2 1 2 1 ...
##  $ Gene_C: Factor w/ 2 levels "0","+1": 2 2 2 2 1 1 1 2 1 1 ...
##  $ Gene_D: Factor w/ 2 levels "0","+1": 1 2 1 1 1 2 2 1 1 1 ...
##  $ Class : Factor w/ 2 levels "normal","tumor": 1 1 2 2 2 1 2 1 2 2 ...

str(test)

## 'data.frame': 1 obs. of  5 variables:
##  $ Gene_A: Factor w/ 1 level "+1": 1
##  $ Gene_B: Factor w/ 1 level "-1": 1
##  $ Gene_C: Factor w/ 1 level "+1": 1
##  $ Gene_D: Factor w/ 1 level "+1": 1
##  $ Class : Factor w/ 1 level "unknown": 1
```

As we can see, the factor levels are not the same in the training and test set. The user has to ensure factor level consistency. A simple solution consists in first appending the test case to the training set and then splitting them apart. Note that the class labels also have to be consistent. At the moment, the test case has the class label "unknown", but this is not a valid label. When we add the test case to the training set, we erroneously increase the factor level of "Class", which in turn will cause `naiveBayes()` to assume that there are three classes in total.

```
train_save <- train
test_save <- test
```

```
X <- rbind(train_save, test_save)
train <- X[1:14, ]
test <- X[15, ]
# Build the model again and apply it to the test case.
NB_model <- naiveBayes(Class ~ ., data = train)
pred <- predict(NB_model, test, type = "raw")
pred

##      normal tumor unknown
## [1,]    NaN   NaN     NaN

#' We have to make sure that our test case has a class label that also appears in
#' the training set, so we can just choose either "normal" or "tumor".
#' It does not matter which one we choose, and it has no influence
#' on predict() because predict() uses only the predictor variables
test$Class <- "normal"
X <- rbind(train_save, test)
train <- X[1:14, ]
test <- X[15, ]
# Build the model again and apply it to the test case.
NB_model <- naiveBayes(Class ~ ., data = train)
pred <- predict(NB_model, test, type = "raw")
pred

##         normal     tumor
## [1,] 0.7954173 0.2045827
```

When we use `naiveBayes()`, we have to make sure that the factor levels in the training and test set are consistent. Also, we need to make sure that data types are correct. Note that the values in Table 3 could be interpreted as integers, which would of course lead to different results (see the R code at `osf.io/gtchm` for more details). Both pitfalls can be easily overlooked and thereby cause `naiveBayes()` and `predict()` to produce results that may look plausible but that are, in fact, incorrect.

4. Discussion

In this article, we derived Bayes' theorem from the fundamental concepts of probability. We then presented one member of the family of machine learning methods that are based on this theorem, the naive Bayes classifier, which is one of the oldest workhorses of machine learning.

It is well known that the misclassification error rate is minimized if each instance is classified as a member of that class for which its conditional class posterior probability is maximal [15]. Consequently, the naive Bayes classifier is optimal (cf. Eq. 21), in the sense that no other classifier is expected to achieve a smaller misclassification error rate, provided that the features are independent. However, this assumption is a rather strong one; clearly, in the vast majority of real-world classification problems, this assumption is violated. This is particularly true for genomic data sets with many co-expressed genes. Perhaps surprisingly, however, the naive Bayes classifier has demonstrated excellent performance even when the data set attributes are not independent [15, 16].

Another advantage of the naive Bayes classifier is that the calculation of the conditional probabilities is highly parallelizable and amenable to distributed processing, for example, in a MapReduce environment [17]. Thus, the naive Bayes classifier is also interesting for big data analytics.

The performance of the naive Bayes classifier can often be improved by eliminating highly correlated features. For example, assume that we add ten additional genes to the data set shown in Table 4, where each gene is described by expression values that are highly correlated to those of Gene B. This means that the estimated conditional probabilities will be dominated by those values, which would "swamp out" the information contained in the remaining genes.

We illustrated some caveats and pitfalls and how to avoid them when building a naive Bayes classifier in the programming language and environment R [14]. Further details with fully commented code and example data are available at the accompanying website `http://osf.io/92mes`.

5. Closing Remarks

Harold Jeffreys, a pioneer of modern statistics, succinctly stated the importance of Bayes' theorem: "[Bayes' theorem] is to the theory of probability what Pythagoras' theorem is to geometry." [18, p.31]. Indeed, Bayes' theorem is of fundamental importance not only for inferential statistics, but also for machine learning, as it underpins the naive Bayes classifier. This classifier has demonstrated excellent performance compared to more sophisticated models in a range of applications, including tumor classification based on gene expression profiling [19]. The naive Bayes classifier performs remarkably well even when the underlying independence assumption is violated.

References

[1] D. Berry, Statistics—A Bayesian Perspective, Duxbury Press, 1996.

[2] D. D. Lewis, Naive (Bayes) at forty: the independence assumption in information retrieval, in: C. Nédellec, C. Rouveirol (Eds.), Machine Learning: ECML-98: 10th European Conference on Machine Learning, Chemnitz, Germany, April 21–23, Springer, Berlin/Heidelberg, 1998, pp. 4–15.

[3] J. Berger, M. Delampady, Testing precise hypotheses, Statistical Science 2 (3) (1987) 317–352.

[4] D. Berrar, Confidence curves: an alternative to null hypothesis significance testing for the comparison of classifiers, Machine Learning 106 (6) (2017) 911–949.

[5] D. Berrar, W. Dubitzky, On the Jeffreys-Lindley Paradox and the looming reproducibility crisis in machine learning, in: Proceedings of the 4th IEEE International Conference on Data Science and Advanced Analytics, Tokyo, Japan, 19–21 October 2017, 2017, pp. 1–7.

[6] T. Sellke, M. Bayarri, J. Berger, Calibration of p values for testing precise null hypotheses, The American Statistician 55 (1) (2001) 62–71.

[7] R. Kass, A. Raftery, Bayes factors, Journal of the American Statistical Association 90 (430) (1995) 773–795.

[8] H. Jeffreys, Theory of Probability, Clarendon Press, Oxford, 3rd edition, reprinted 2003, Appendix B, p.432, 1961.

[9] D. Berrar, An empirical evaluation of ranking measures with respect to robustness to noise, Journal of Artificial Intelligence Research 49 (2014) 241–267.

[10] D. Berrar, P. Flach, Caveats and pitfalls of ROC analysis in clinical microarray research (and how to avoid them), Briefings in Bioinformatics 13 (1) (2012) 83–97.

[11] G. Gigerenzer, W. Gaissmaier, E. Kurz-Milcke, L. Schwartz, S. Woloshin, Helping doctors and patients to make sense of health statistics, Psychological Science in the Public Interest 8 (2) (2008) 53–96.

[12] I. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, 2nd edition, 2005.

[13] D. Meyer, E. Dimitriadou, K. Hornik, A. Weingessel, F. Leisch, e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien, R package version 1.6-7 (2015).
URL `https://CRAN.R-project.org/package=e1071`

[14] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria (2017).
URL `https://www.R-project.org/`

[15] P. Domingos, M. Pazzani, On the optimality of the simple Bayesian classifier under zero-one loss, Machine Learning 29 (2) (1997) 103–130.

[16] N. A. Zaidi, J. Cerquides, M. J. Carman, G. I. Webb, Alleviating naive Bayes attribute independence assumption by attribute weighting, Journal of Machine Learning Research 14 (2013) 1947–1988.

[17] S. Villa, M. Rossetti, Learning continuous time Bayesian network classifiers using MapReduce, Journal of Statistical Software 62 (3) (2014) 1–25.

[18] H. Jeffreys, Scientific Inference, Cambridge University Press, 3rd edition, 1973.

[19] S. Dudoit, J. Fridlyand, T. Speed, Comparison of discrimination methods for the classification of tumors using gene expression data, Journal of the American Statistical Association 97 (457) (2002) 77–87.