Naive Bayes

Hiroshi Shimodaira∗
6 February 2015

In the previous chapter we introduced the use of Bayes’ Theorem for pattern classification. For a test
vector \( x \), we estimate the posterior probability \( P(c_k | x) \) for each class \( c_k \). To classify \( x \) we choose the
class with the largest estimated posterior probability. To do this we re-express the posterior probabilities
using Bayes’ Theorem:

\[
P(c_k | x) = \frac{P(x | c_k) P(c_k)}{P(x)} \propto P(x | c_k) P(c_k).
\]

Thus for each class we need to provide an estimate of the likelihood \( P(x | c_k) \) and the prior \( P(c_k) \).

To estimate the likelihood we need a statistical model which can provide a likelihood estimate. In the
“fish” example, the likelihood was estimated using a histogram for each class: the likelihood of a fish
with length \( x \) coming from class \( c_k \) was estimated as the relative frequency of fish of length \( x \) estimated
from the training data for class \( c_k \).

Imagine if in addition to the length we also had some other information such as weight and circum-
ference. In this case, the input feature vector for each fish would contain 3 elements: (length, weight, circumference). If we have 20 possible values for each feature, then the number of bins in a histogram of these vectors would be \( 20^3 = 8000 \). In this case 100 examples per class would mean that it is not possible to observe examples for the vast majority of the bins. We would need many more examples to obtain reasonable estimates based on relative frequencies. As the number of feature dimensions increases, the total number of possible feature vectors increases exponentially.

This severe difficulty that arises from moving to spaces of higher dimension was termed the curse of
dimensionality by Richard Bellman in the 1950s. This is a critical problem: it becomes impossible to
reliably estimate histograms once we have more than a few dimensions, even if we have millions of
examples.

In this chapter we shall look at an approach to the problem called the Naive Bayes approximation.
After showing how it works in a small, fictitious example, we’ll go on to see how this approach may be
applied to the problem of text classification, which uses high-dimensional feature vectors (e.g., \( 10^4 \)-to
\( 10^6 \)-dimensional).

1 The Naive Bayes assumption

One way to deal with the curse of dimensionality is to assume that the different feature dimensions
are independent. If we are using histograms of relative frequencies to estimate likelihoods, then this

means that we construct, for each class, \( d \) 1-dimensional histograms, rather than a single \( d \)-dimensional
histogram. If we assume that each dimension can take \( m \) values, we now only need to estimate \( dm \)
relative frequencies, rather than \( m^d \) relative frequencies!

Consider a \( d \)-dimensional feature vector \( x = (x_1, x_2, \ldots, x_d) \). We write the probability of a data point \( x \)
given class \( c_k \) as a joint distribution of the \( d \) components of \( x \) (equation 1). We can re-express any joint
distribution as a product of conditional distributions using successive applications of the product rule
(equation 2). The result is known as the chain rule of probability, which in this case is:

\[
P(x | c_k) = P(x_1 | x_2, x_3, \ldots, x_d | c_k) P(x_2 | x_3, \ldots, x_d | c_k) \cdots P(x_d | c_k).
\]

This decomposition in itself doesn’t address the curse of dimensionality, since the first term on the
right-hand side of (2) is conditioned on \( (d - 1) \) terms, the second on \( (d - 2) \) terms, and so on.

However we can simplify things if we naïvely assume that the individual feature dimensions \((x_1, x_2, \ldots, x_d)\)
are independent, that is:

\[
P(x_1 | x_2, \ldots, x_d, c_k) = P(x_1 | c_k) \quad P(x_2 | x_3, \ldots, x_d, c_k) = P(x_2 | c_k) \quad \ldots
\]

This is called the Naive Bayes assumption. The assumption is drastic and rarely true: for example,
in the above case Naive Bayes states that the length of a male fish is independent of its weight and
circumference. However making this approximation allows us to have a much simpler form for the
likelihood, since (2) is simplified to:

\[
P(x | c_k) = P(x_1 | c_k) P(x_2 | c_k) \cdots P(x_d | c_k) = \prod_{i=1}^{d} P(x_i | c_k).
\]

We have approximated the probability of a \( d \)-dimensional feature vector as a product of \( d \) probabilities
of the 1-dimensional feature vectors.

Using this assumption, we can express Bayes’ theorem as follows:

\[
P(c_k | x) = \frac{P(x | c_k) P(c_k)}{P(x)} \propto \prod_{i=1}^{d} P(x_i | c_k) P(c_k)
\]

2 Example

The following (fictitious) example comes from the book Data Mining by Witten and Frank.

Consider a game which is played or not depending on the weather conditions: outlook (sunny / overcast /
rainy), temperature (hot / mild / cool), humidity (high / normal) and windy (true / false). The input
variable is 4-dimensional, with 2 or 3 values per dimension. There are 36 possible combinations
(3 \times 3 \times 2 \times 2).

We have an input training set of 14 examples, shown in table 1. As is usually the case in machine
learning, there are fewer training examples than possible settings of the input variables; most possible
Learning and Data Note 6 Informatics 2B

Outlook Temperature Humidity Windy | Play
--- | --- | --- | --- | ---
sunny | hot | high | false | NO
sunny | hot | high | true | NO
overcast | hot | high | false | YES
rainy | mild | high | false | YES
rainy | cool | normal | false | YES
rainy | cool | normal | true | NO
overcast | cool | normal | true | YES
overcast | hot | normal | false | NO
rainy | mild | high | true | YES
overcast | hot | normal | false | YES
rainy | mild | high | true | NO

Table 1: Training data for the weather example

Outlook Y N Temperature Y N Humidity Y N Windy Y N
--- | --- | --- | --- | --- | --- | --- | --- | --- | ---
sunny | 2 | 3 | hot | 2 | 2 | high | 3 | 4 | false | 6 | 2
overcast | 4 | 0 | mild | 4 | 2 | normal | 6 | 1 | true | 3 | 3
rainy | 3 | 2 | cool | 3 | 1 | normal | 6 | 1 | true | 3 | 3

Table 2: Play counts for different weather conditions.

Outlook Y N Temperature Y N Humidity Y N Windy Y N
--- | --- | --- | --- | --- | --- | --- | --- | --- | ---
sunny | 2/9 | 3/5 | hot | 2/9 | 2/5 | high | 3/9 | 4/5 | false | 6/9 | 2/5
overcast | 4/9 | 0/5 | mild | 4/9 | 2/5 | normal | 6/9 | 1/5 | true | 3/9 | 3/5
rainy | 3/9 | 2/5 | cool | 3/9 | 1/5 | normal | 6/9 | 1/5 | true | 3/9 | 3/5

Table 3: Play relative frequencies for different weather conditions. There was play on 9/14 cases.

conditions are not directly observed. We can tabulate the data in terms of the frequencies for each of the four input variables (table 2), and in terms of the relative frequencies (table 3). The relative frequencies can be used as probability estimates, for example \( P(T = h \mid \text{Play} = Y) = 2/9 \).

We are given the following test example:

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Temp.</th>
<th>Humidity</th>
<th>Windy</th>
<th>Play</th>
</tr>
</thead>
</table>
sunny | cool | high | true | ? |

This example’s feature vector, \( x \), was not observed in the training set. We can generalise from the other examples by using Naive Bayes:

\[
P(\text{play} = Y \mid x) \propto P(\text{play} = Y) \cdot P(O = s \mid \text{play} = Y) \cdot P(T = c \mid \text{play} = Y) \\
\cdot P(W = t \mid \text{play} = Y)
\]

\[
= \frac{9}{14} \cdot \frac{2}{9} \cdot \frac{3}{5} \cdot \frac{3}{5} = 0.0053
\]

\[
P(\text{play} = N \mid x) \propto P(\text{play} = N) \cdot P(O = s \mid \text{play} = N) \cdot P(T = c \mid \text{play} = N) \\
\cdot P(W = t \mid \text{play} = N)
\]

\[
= \frac{5}{14} \cdot \frac{3}{5} \cdot \frac{1}{5} \cdot \frac{4}{5} = 0.0206
\]

And the ratio of posterior probabilities is:

\[
\frac{P(\text{play} = Y \mid x)}{P(\text{play} = N \mid x)} = \frac{0.0053}{0.0206} = 0.26
\]

Hence we classify \( x \) as \( \text{play} = N \).

3 Conclusion

In this chapter we have introduced the Naive Bayes approximation. When we have a multidimensional feature vector, Naive Bayes approximated the likelihoods by considering each feature dimension to be independent:

\[
P(x_1, x_2, x_3 \mid C) \approx P(x_1 \mid C) \cdot P(x_2 \mid C) \cdot P(x_3 \mid C),
\]

which approximates a \( d \)-dimensional distribution as \( d \) 1-dimensional distributions. If each dimension can take \( m \) different values, we need only estimate \( md \) probabilities, rather than \( m^d \) probabilities.