Learning: Naïve Bayes Classifiers

CSE 415: Introduction to Artificial Intelligence
University of Washington
Spring, 2017

© S. Tanimoto and University of Washington, 2017

Outline

• Motivation
• A discrete example: Classifying fruits
• The Naïve Bayes assumption
• Maximum likelihood estimation of probabilities from samples
• A continuous example using a Gaussian model: Classifying online shoppers.

Motivation

• Bayes’ rule is a general technique in classification, but costly in terms of requiring large training sets.
• By making independence assumptions, much less training data is required.
• Often the results are very good.
• Naïve Bayes classifiers are based on the assumption that likelihoods of each feature are independent of those of other features.

A Discrete Example

A training example consists of a vector of attribute values with a category indicator.

Example Training Data Stats

<table>
<thead>
<tr>
<th>Long</th>
<th>Yellow</th>
<th>class</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>No</td>
<td>Lemon</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>Banana</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>Other</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Lemon</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Banana</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Other</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Lemon</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Banana</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Other</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Lemon</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Banana</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Other</td>
</tr>
</tbody>
</table>

Compute the priors:
P(lemon) = 7 / 25
P(banana) = 12 / 25
P(other) = 6 / 25
P(long) = 14 / 25
P(yellow) = 15 / 25

Compute the likelihoods of individual features:
P(long | lemon) = 0
P(long | banana) = 12 / 12
P(long | other) = 2 / 6
P(yellow | lemon) = 5 / 7
P(yellow | banana) = 9 / 12
P(yellow | other) = 1 / 6
To Classify An Instance

Let’s call the vector “(not long, yellow), ?” the evidence E. Ideally, we would get the a posteriori probability of each class and choose the class with the highest:

\[ P(\text{lemon} \mid E), \ P(\text{banana} \mid E), \ P(\text{other} \mid E). \]

This would require applying Bayes’ rule as follows, e.g., for lemon:

\[ P(\text{lemon} \mid E) = \frac{P(E \mid \text{lemon}) P(\text{lemon})}{P(E)} \]

However, we don’t have \( P(E \mid \text{lemon}) \) in the feature likelihoods we calculated.

We could, in principle, compute \( P(E \mid \text{lemon}) \) using this:

\[ P(\text{not long, yellow} \mid \text{lemon}) = P(\text{not long} \mid \text{lemon and yellow}) P(\text{yellow} \mid \text{lemon}). \]

But we don’t have the first of these readily available, either.

It’s a lot easier if we assume that \( P(\text{long} \mid \text{lemon}) \) and \( P(\text{yellow} \mid \text{lemon}) \) are independent. Then we can approximate \( P(E \mid \text{lemon}) \) as

\[ P(\text{not long and yellow} \mid \text{lemon}) \approx (1 - P(\text{long} \mid \text{lemon})) P(\text{yellow} \mid \text{lemon}). \]

Classification with N.B.

\[ P(\text{lemon} \mid E) \approx \frac{(1 - P(\text{long} \mid \text{lemon})) P(\text{yellow} \mid \text{lemon}) P(\text{lemon})}{P(E)} \]

\[ = \frac{(1 - 0) (5/7) (7/25)}{P(E)} = \frac{35/185}{P(E)} = \frac{1}{5} / P(E) \]

\[ P(\text{banana} \mid E) \approx \frac{(1 - P(\text{long} \mid \text{banana})) P(\text{yellow} \mid \text{banana}) P(\text{banana})}{P(E)} \]

\[ = \frac{(1 - 12/12) (9/12) (12/25)}{P(E)} = 0 \]

\[ P(\text{other} \mid E) \approx \frac{(1 - P(\text{long} \mid \text{other})) P(\text{yellow} \mid \text{other}) P(\text{other})}{P(E)} \]

\[ = \frac{(1 - 2/6) (1/6) (6/25)}{P(E)} = \frac{2/75}{P(E)} \]

Clearly \( P(\text{lemon} \mid E) \) gets a higher value than \( P(\text{other} \mid E) \), and so the new instance is classified as a lemon (assuming we are using the Maximum A Posteriori (MAP) probability classification rule).

Exercise

Using the same Naive Bayes classifier,

\[ \text{Classify:} \ (\not\text{not long, not yellow}, ?) \]

1. What is \( P(\text{lemon} \mid E) \)?
2. What is \( P(\text{banana} \mid E) \)?
3. What is \( P(\text{other} \mid E) \)?

What is \( \text{argmax}_f P(f \mid E) \)?

A Difficulty When Using Frequencies for Probabilities

Small training sets tend to lead to some zeros in the counts, even when the underlying distributions have nonzero probabilities. These zeros can wreak havoc with Naive Bayes classifiers. Therefore ...

Using Maximum Likelihood to Estimate the Probability Distribution

Instead of taking the count ratios as probabilities, use them as evidence for underlying distributions and estimate those distributions.

A set of distributions can be parameterized by \( \theta \). So choose the best \( \theta \): the one that has the highest likelihood given the training data.

\[ \theta_{\text{best}} \leftarrow \text{argmax}_\theta P(\theta \mid T) \]

We can estimate the parameters of \( \theta \) by including phantom examples in our training set to make sure that no count is 0. By adding 1 to every count, we get frequency ratios that generally do not add up to 1 any more, and so we normalize the new ratios so they DO add up to 1.
Naïve Bayes Classifiers with Continuous-Valued Features

Let $X = (x_0, x_1, ..., x_{n-1})$ be a vector in $\mathbb{R}^n$.

We can still use Bayes’ rule to compute posterior values useful for classification. However, the values will be probability density values rather than probabilities.

$$P(y | X) = \frac{P(X | y)P(y)}{P(X)}$$

The Naïve Bayes assumption of conditional independence is

$$P(X | y) = P(x_0 | y)P(x_1 | y) \cdots P(x_{n-1} | y)$$

Continuous Features with Gaussian Distributions

Let’s assume each $x_i$ in $(x_0, x_1, ..., x_{n-1})$ comes from a probability distribution given by the probability density function (pdf): $P(x_i = x | y=c) = \exp((x - \mu_{i,c})^2 / 4\sigma_{i,c}^2)$

The Naïve Bayes assumption is

$$P(X | y=c) = \exp((x - \mu_{0,c})^2 / 4\sigma_{0,c}^2) \cdots \exp((x - \mu_{n-1,c})^2 / 4\sigma_{n-1,c}^2)$$

Take the logarithm of both sides:

$$\ln P(X | y=c) = (x - \mu_{0,c})^2 / 4\sigma_{0,c}^2 + \cdots + (x - \mu_{n-1,c})^2 / 4\sigma_{n-1,c}^2$$

Example of N.B. with Gaussians

Consider the case of the Northwest Hiking Supply Company, trying to increase their online sales. When a potential customer visits their website, they want to be able to predict, after 1 minute, whether that visitor is a hot prospect, so that they can offer a special “closer” promotion. For training purposes, any past visitor who has purchased something during the visit is considered “hot.”

By the end of 60 seconds at the website, they have two feature values: “dwell time” ($D$) and “number of product revisits” ($R$). The range of values of $D$ is 0 to 60, and for $R$ it is 0 to 10. From these two measures, they want to classify the visitor as “hot” (H) or “not” (N).

Naïve Bayes Classifiers require only a relatively small amount of training examples (linear in the number of features times number of values, in the discrete case).

Whereas the full joint distribution typically requires $\Omega(2^n)$ training examples, which is intractable when $n$ is large (e.g., $n > 50$).

Naïve Bayes classification is fast.

The Naïve Bayes assumption of conditional independence, while not usually an accurate model of the underlying joint distribution, still works remarkably well.