Estimating class prevalences $P(y)$

How to estimate $P(y)$?

Simplest estimate: For each class, just divide the empirical frequency by the total number of examples:

$$P(y = \alpha) = \frac{\#y = \alpha}{N}$$  \hspace{1cm} (10)

For example: The proportion of females in a computer science class at our department is estimated by the number of females in this class divided by the size of the class. (This is an unbiased estimate if the students taking the course is a random sample of the students at the department... this may or may not be true.)

In the i.i.d. setting it is consistent, and it works reasonably well when there are a sufficient number of examples of all classes.
Estimating class-conditional distributions $P(x \mid y)$

- How to estimate $P(x \mid y)$ for each value of $y$?
  - For high-dimensional $x$ this is an extremely non-trivial problem
  - Discrete $x$:
    Dividing empirical frequencies by total count for every element of $\mathcal{X}$ does not work in practice in most cases (too many states, too few data points $\Rightarrow$ unreliable, many zeros)
  - Continuous-valued $x$:
    Multivariate normal (Gaussian) distribution ($n$ dimensions):
    $$
    \mathcal{N}(x; \mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp \left( -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right)
    $$
    ‘Only’ requires estimates of the mean $\mu$ and the covariance matrix $\Sigma$, so the total number of parameters is $n + n(n + 1)/2$. For low-dimensional data this may be ok, but even this is unreliable for high-dimensional data.
Naïve Bayes classifier

The central idea of the Naïve Bayes classifier is to assume that the class-conditional distributions factorize, i.e. that for each class, the joint distribution over examples is given by the product of distributions in each dimension:

$$P(x \mid y = y_0) = \prod_{i=1}^{n} P(x_i \mid y = y_0)$$  \hspace{1cm} (11)

In other words, one estimates the distributions of each class-attribute pair separately and uses these to approximate the class-conditional distributions in the joint space.
Naïve Bayes classifier – example 1

- Gaussian class-conditional distributions:
  - Instead of $n + n(n+1)/2$ parameters for each class-conditional distribution, we only use $n + n$ (mean and variance for each dimension)
  - Fit to data is only approximate, especially when there are strong correlations between attributes within one or more classes

\[
p(x \mid y = 0)\]

\[
p(x \mid y = 1)\]

true class-conditional distributions

Naive Bayes approximation (note: axis-aligned!)

\[
\prod_i p(x_i \mid y = 0)\]

\[
\prod_i p(x_i \mid y = 1)\]
Naïve Bayes classifier – example 2

Discrete attributes ($n$ attributes, each with $L$ possible values):

- Instead of $L^n - 1$ parameters for each class-conditional distribution (for the joint distribution), we only need $(L - 1)n$ parameters (for the $n$ marginal distributions each over $L$ values).

\[
p(x \mid y = 1) = \prod_{i} p(x_i \mid y = 1)
\]
\[
p(x \mid y = 0) = \prod_{i} p(x_i \mid y = 0)
\]

true class-conditional distributions

Naive Bayes approximation
Naïve Bayes – fitting univariate class conditionals

- Continuous variables: densities (see Appendix C)
  - Gaussian
  - Exponential
  - Student’s t
  - Histogram (i.e. discretizing)
  - Kernel density estimation
  - …

- How to estimate the parameters?
  - Moment matching
  - Maximum likelihood
  - …
Naïve Bayes – fitting univariate class conditionals

- Discrete variables with finite state space:
  - We already gave a simple estimator when discussing estimating $P(y)$: just normalizing the counts. In this case, for a given class $y = y_0$, and a given attribute $x_i$ we would normalize the counts by
    \[
P(x_i = \alpha \mid y = y_0) = \frac{\#[x_i = \alpha \text{ and } y = y_0]}{\#[y = y_0]} \tag{12}\]
  - For some class-attribute pairs, some of the counts may be zero
    ⇒ some of the class-conditional probabilities are zero!
    ⇒ for some (new) data points $x$, we have $\forall y : P(x \mid y) = 0$, so we cannot compute $P(y \mid x)$ (it is undefined!)
  - A simple solution: Add a constant $c$ (for example $c = 1$) to the numerator, and add $Lc$ (where $L$ is the number of distinct values of $x_i$) to the denominator ⇒ no more zeros, probabilities still sum to one, and can even be justified on theoretical grounds
Example (spam filtering):

<table>
<thead>
<tr>
<th></th>
<th>viagra</th>
<th>millions</th>
<th>grade</th>
<th>confidential</th>
</tr>
</thead>
<tbody>
<tr>
<td>msg #1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>msg #2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>msg #3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>msg #4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>msg #5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>spam #1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>spam #2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>spam #3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>spam #4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>spam #5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Simply normalizing the frequencies:

\[
P(\text{msg}) = \frac{5}{10} = 0.5
\]
\[
P(\text{spam}) = \frac{5}{10} = 0.5
\]
\[
P(\text{viagra} | \text{msg}) = \frac{0}{5} = 0
\]
\[
P(\text{millions} | \text{msg}) = \frac{2}{5} = 0.4
\]
\[
P(\text{grade} | \text{msg}) = \frac{3}{5} = 0.6
\]
\[
P(\text{confidential} | \text{msg}) = \frac{2}{5} = 0.4
\]
\[
P(\text{viagra} | \text{spam}) = \frac{2}{5} = 0.4
\]
\[
P(\text{millions} | \text{spam}) = \frac{3}{5} = 0.6
\]
\[
P(\text{grade} | \text{spam}) = \frac{0}{5} = 0
\]
\[
P(\text{confidential} | \text{spam}) = \frac{3}{5} = 0.6
\]

Compute: \( P(\text{spam} | \text{“confidential: how to win millions”}) \) (next slide)

Problem: A new message “premium grade viagra available now!!!” would get 0 probability under both classes!
email = "confidential: how to win millions"

\[
P(\text{spam} \mid \text{email}) = P(\text{email} \mid \text{spam})P(\text{spam})/P(\text{email})
\]
\[
= P(\neg \text{viagra} \mid \text{spam})P(\text{millions} \mid \text{spam})P(\neg \text{grade} \mid \text{spam})P(\text{confidential} \mid \text{spam})P(\text{spam})/P(\text{email})
\]
\[
= 0.6 \times 0.6 \times 1.0 \times 0.6 \times 0.5/P(\text{email}) = 0.108/P(\text{email})
\]

\[
P(\text{msg} \mid \text{email}) = P(\text{email} \mid \text{msg})P(\text{msg})/P(\text{email})
\]
\[
= P(\neg \text{viagra} \mid \text{msg})P(\text{millions} \mid \text{msg})P(\neg \text{grade} \mid \text{msg})P(\text{confidential} \mid \text{msg})P(\text{msg})/P(\text{email})
\]
\[
= 1.0 \times 0.4 \times 0.4 \times 0.4 \times 0.5/P(\text{email}) = 0.032/P(\text{email})
\]

\[
P(\text{email}) = 0.108 + 0.032 = 0.140
\]

\[
P(\text{spam} \mid \text{email}) = 0.108/0.140 \approx 0.77
\]
\[
P(\text{msg} \mid \text{email}) = 0.032/0.140 \approx 0.23
\]
Example (spam filtering):

Instead, let's add $c = 1$ to all counts:

\[
P(\text{msg}) = \frac{5 + 1}{12} = 0.5
\]
\[
P(\text{spam}) = \frac{5 + 1}{12} = 0.5
\]

\[
P(\text{viagra} \mid \text{msg}) = \frac{0 + 1}{7} = \frac{1}{7}
\]
\[
P(\text{millions} \mid \text{msg}) = \frac{2 + 1}{7} = \frac{3}{7}
\]
\[
P(\text{grade} \mid \text{msg}) = \frac{3 + 1}{7} = \frac{4}{7}
\]
\[
P(\text{confidential} \mid \text{msg}) = \frac{2 + 1}{7} = \frac{3}{7}
\]
\[
P(\text{viagra} \mid \text{spam}) = \frac{2 + 1}{7} = \frac{3}{7}
\]
\[
P(\text{millions} \mid \text{spam}) = \frac{3 + 1}{7} = \frac{4}{7}
\]
\[
P(\text{grade} \mid \text{spam}) = \frac{0 + 1}{7} = \frac{1}{7}
\]
\[
P(\text{confidential} \mid \text{spam}) = \frac{3 + 1}{7} = \frac{4}{7}
\]

Note: Adding 2 to the denominator because each variable is binary (has 2 states). Now all probabilities are $> 0$. 
Practical issues with Naïve Bayes

- In high dimensions, the probabilities/densities $P(x \mid y_j)$ tend to become very close to zero for all classes $y_j$. The same applies to the marginals $P(x)$.

⇒ Numerical problems in comparing the $P(x \mid y_j)$ with each other, and in computing $P(x \mid y_j)P(y_j)/P(x)$ (floating point underflow, 0/0 problems)

- Solution: Use logarithms and add a suitable constant before exponentiating, as follows

1. $L(y_j \mid x) = \log(P(x \mid y_j)P(y_j)) = \sum_i \log(P(x_i \mid y_j)) + \log(P(y_j))$
2. $L(y_j \mid x) = L(y_j \mid x) - \max_j(L(y_j \mid x))$
3. $P_t(y_j \mid x) = \exp(L(y_j \mid x))$
4. $P(y_j \mid x) = P_t(y_j \mid x)/\sum_j P_t(y_j \mid x)$
Generative vs discriminative

- Naïve Bayes is a representative of what we call a *generative* approach to classification...
  - From the model, it is actually possible to synthesize (generate) new examples: First draw $y \sim P(y)$ then draw $x \sim P(x \mid y)$

- ...as opposed to a *discriminative* approach, which only models $P(y \mid x)$ or even more simply just the function $\hat{y} = f(x)$

- Note: The generative approach is, in a sense, solving a more difficult problem than is asked for.