Learning Algorithms: Naive Bayes
Homework #6 has been released. It is due at 11:59PM on 11/7.
Group Project Updates are due on 11/12 at 11:59PM
Group Project Presentation Guidelines Have Been Posted
What does a simple supervised learning method look like?
Naive Bayes

Assume all features are independent effects of the label

![Diagram showing Y connected to F₁, F₂, ..., Fₙ]

Good assumption?
Naive Bayes for Digits

Simple digit recognition version:

- One feature (variable) $F_{i,j}$ for each grid position $<i,j>$
- Feature values are on / off, based on whether intensity is more or less than 0.5 in underlying image
- Each input maps to a feature vector, e.g.

\[
1 \rightarrow \langle F_{0,0} = 0 \ F_{0,1} = 0 \ F_{0,2} = 1 \ F_{0,3} = 1 \ F_{0,4} = 0 \ \ldots \ F_{15,15} = 0 \rangle
\]

- Here: lots of features, each is binary valued

Naive Bayes model: 

\[
P(Y|F_{0,0} \ldots F_{15,15}) \propto P(Y) \prod_{i,j} P(F_{i,j}|Y)
\]

What do we need to learn?
General Naive Bayes

A general Naive Bayes model:

\[ P(Y, F_1 \ldots F_n) = P(Y) \prod_i P(F_i|Y) \]

\(|Y|\) parameters

\(|Y| \times |F|^n\) values

\(n \times |F| \times |Y|\) parameters

We only have to specify how each feature depends on the class
Total number of parameters is \textbf{linear} in \(n\)
Model is very simplistic, but often works anyway
Inference for Naive Bayes

Goal: compute posterior distribution over label variable $Y$

Step 1: get joint probability of label and evidence for each label

$$P(Y, f_1 \ldots f_n) = \begin{bmatrix} P(y_1, f_1 \ldots f_n) \\ P(y_2, f_1 \ldots f_n) \\ \vdots \\ P(y_k, f_1 \ldots f_n) \end{bmatrix}$$

Step 2: sum to get probability of evidence

$$P(f_1 \ldots f_n) = P(y_1) \prod_i P(f_i|y_1) + P(y_2) \prod_i P(f_i|y_2) + \cdots + P(y_k) \prod_i P(f_i|y_k)$$

Step 3: normalize by dividing Step 1 by Step 2

$$P(Y|f_1 \ldots f_n)$$
What do we need in order to use Naive Bayes?

Inference method (we just saw this part)

- Start with a bunch of probabilities: $P(Y)$ and the $P(F_i|Y)$ tables
- Use standard inference to compute $P(Y|F_1\ldots F_n)$
- Nothing new here
General Naive Bayes

Estimates of local conditional probability tables

- \( P(Y) \), the prior over labels
- \( P(F_i|Y) \) for each feature (evidence variable)
- These probabilities are collectively called the **parameters** of the model and denoted by \( \Theta \)

- Up until now, we assumed these appeared by magic, but...
- …they typically come from training data counts: we’ll look at this soon
Example: Conditional Probabilities

\[ P(Y) \]

\[
\begin{array}{c|c}
1 & 0.1 \\
2 & 0.1 \\
3 & 0.1 \\
4 & 0.1 \\
5 & 0.1 \\
6 & 0.1 \\
7 & 0.1 \\
8 & 0.1 \\
9 & 0.1 \\
0 & 0.1 \\
\end{array}
\]

\[ P(F_{3,1} = \text{on}|Y) \]

\[
\begin{array}{c|c}
1 & 0.01 \\
2 & 0.05 \\
3 & 0.05 \\
4 & 0.30 \\
5 & 0.80 \\
6 & 0.90 \\
7 & 0.05 \\
8 & 0.60 \\
9 & 0.50 \\
0 & 0.80 \\
\end{array}
\]

\[ P(F_{5,5} = \text{on}|Y) \]

\[
\begin{array}{c|c}
1 & 0.05 \\
2 & 0.01 \\
3 & 0.90 \\
4 & 0.80 \\
5 & 0.90 \\
6 & 0.90 \\
7 & 0.25 \\
8 & 0.85 \\
9 & 0.60 \\
0 & 0.80 \\
\end{array}
\]
Naive Bayes for Text

Bag-of-words Naive Bayes:

- Features: $W_i$ is the word at position $i$
- As before: predict label conditioned on feature variables (spam vs. ham)
- As before: assume features are conditionally independent given label
- New: each $W_i$ is identically distributed

Generative model: $P(Y, W_1 \ldots W_n) = P(Y) \prod_{i} P(W_i | Y)$
Naive Bayes for Text

“Tied” distributions and bag-of-words

- Usually, each variable gets its own conditional probability distribution $P(F \mid Y)$
- In a bag-of-words model
  - Each position is identically distributed
  - All positions share the same conditional probs. $P(W \mid Y)$
- Why make this assumption?
- Called “bag-of-words” because model is insensitive to word order or reordering
Example: Spam Filtering

Model: \( P(Y, W_1 \ldots W_n) = P(Y) \prod_i P(W_i|Y) \)

What are the parameters?

\[
\begin{align*}
P(Y) & \quad P(W|\text{spam}) & \quad P(W|\text{ham})
\end{align*}
\]

Where do these tables come from?

Slide credit: Dan Klein and Pieter Abbeel, UC Berkeley CS 188
Spam Example

| Word     | P(w|spam) | P(w|ham) | Tot Spam | Tot Ham |
|----------|----------|---------|----------|---------|
| (prior)  | 0.33333  | 0.66666 | -1.1     | -0.4    |

Slide credit: Dan Klein and Pieter Abbeel, UC Berkeley CS 188
Parameter Estimation
Parameter Estimation

• Estimating the distribution of a random variable
• Elicitation: ask a human (why is this hard?)
• Empirically: use training data (learning!)
  ▸ e.g.,: for each outcome $x$, look at the empirical rate of that value:

$$P_{ML}(x) = \frac{\text{count}(x)}{\text{total samples}}$$

$$P_{ML}(r) = \frac{2}{3}$$

• This is the estimate that maximizes the likelihood of the data

$$L(x, \theta) = \prod_{i} P_{\theta}(x_i)$$
Smoothing
Maximum Likelihood

Relative frequencies are the maximum likelihood estimates

\[ \theta_{ML} = \arg \max_{\theta} P(X|\theta) \]
\[ = \arg \max_{\theta} \prod_{i} P_{\theta}(X_{i}) \]
\[ \Rightarrow P_{ML}(x) = \frac{\text{count}(x)}{\text{total samples}} \]

Another option is to consider the most likely parameter value given the data

\[ \theta_{MAP} = \arg \max_{\theta} P(\theta|X) \]
\[ = \arg \max_{\theta} P(X|\theta)P(\theta)/P(X) \]
\[ = \arg \max_{\theta} P(X|\theta)P(\theta) \]
Unseen Events
Laplace Smoothing

Laplace’s estimate:

- Pretend you saw every outcome once more than you actually did

\[ P_{LAP}(x) = \frac{c(x) + 1}{\sum_x [c(x) + 1]} \]

\[ = \frac{c(x) + 1}{N + |X|} \]

- Can derive this estimate with Dirichlet priors
Laplace Smoothing

Laplace’s estimate (extended):

- Pretend you saw every outcome $k$ extra times

$$P_{LAP,k}(x) = \frac{c(x) + k}{N + k|X|}$$

- What’s Laplace with $k = 0$?
- $k$ is the strength of the prior

Laplace for conditionals:

- Smooth each condition independently:

$$P_{LAP,k}(x|y) = \frac{c(x, y) + k}{c(y) + k|X|}$$

$P_{LAP,0}(X) = \quad P_{LAP,1}(X) = \quad P_{LAP,100}(X) =$

Slide credit: Dan Klein and Pieter Abbeel, UC Berkeley CS 188
Estimation: Linear Interpolation

In practice, Laplace often performs poorly for $P(X \mid Y)$:
- When $|X|$ is very large
- When $|Y|$ is very large

Another option: linear interpolation
- Also get the empirical $P(X)$ from the data
- Make sure the estimate of $P(X \mid Y)$ isn’t too different from the empirical $P(X)$

$$P_{LIN}(x \mid y) = \alpha \hat{P}(x \mid y) + (1.0 - \alpha) \hat{P}(x)$$

What if $\alpha$ is 0? 1?
Real NB: Smoothing

For real classification problems, smoothing is critical

New odds ratios:

\[
\frac{P(W|\text{ham})}{P(W|\text{spam})}
\]

<table>
<thead>
<tr>
<th>Word</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>helvetica</td>
<td>11.4</td>
</tr>
<tr>
<td>seems</td>
<td>10.8</td>
</tr>
<tr>
<td>group</td>
<td>10.2</td>
</tr>
<tr>
<td>ago</td>
<td>8.4</td>
</tr>
<tr>
<td>areas</td>
<td>8.3</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

\[
\frac{P(W|\text{spam})}{P(W|\text{ham})}
\]

<table>
<thead>
<tr>
<th>Word</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>verdana</td>
<td>28.8</td>
</tr>
<tr>
<td>Credit</td>
<td>28.4</td>
</tr>
<tr>
<td>ORDER</td>
<td>27.2</td>
</tr>
<tr>
<td>&lt;FONT&gt;</td>
<td>26.9</td>
</tr>
<tr>
<td>money</td>
<td>26.5</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

Do these make more sense?
Errors, and what to do

Examples of errors

Dear GlobalSCAPE Customer,

GlobalSCAPE has partnered with ScanSoft to offer you the latest version of OmniPage Pro, for just $99.99* - the regular list price is $499! The most common question we've received about this offer is - Is this genuine? We would like to assure you that this offer is authorized by ScanSoft, is genuine and valid. You can get the . . .

. . . To receive your $30 Amazon.com promotional certificate, click through to

   http://www.amazon.com/apparel

and see the prominent link for the $30 offer. All details are there. We hope you enjoyed receiving this message. However, if you'd rather not receive future e-mails announcing new store launches, please click . . .

Slide credit: Dan Klein and Pieter Abbeel, UC Berkeley CS 188
What to do about errors?

Need more features—words aren’t enough!

- Have you emailed the sender before?
- Have 1K other people just gotten the same email?
- Is the sending information consistent?
- Is the email in ALL CAPS?
- Do inline URLs point where they say they point?
- Does the email address you by (your) name?

Can add these information sources as new variables in the NB model

Slide credit: Dan Klein and Pieter Abbeel, UC Berkeley CS 188
Baselines

First step: get a baseline

- Baselines are very simple “straw man” procedures
- Help determine how hard the task is
- Help know what a “good” accuracy is
Baselines

Weak baseline: most frequent label classifier

- Gives all test instances whatever label was most common in the training set
- e.g., for spam filtering, might label everything as ham
- Accuracy might be very high if the problem is skewed
- e.g., calling everything “ham” gets 66%, so a classifier that gets 70% isn’t very good…

For real research, usually use previous work as a (strong) baseline
The **confidence** of a probabilistic classifier:

- Posterior over the top label

\[
\text{confidence}(x) = \max_y P(y|x)
\]

- Represents how sure the classifier is of the classification
- Any probabilistic model will have confidences
- No guarantee confidence is correct
Confidences from a Classifier

Calibration

- Weak calibration: higher confidences mean higher accuracy
- Strong calibration: confidence predicts accuracy rate
- What’s the value of calibration?
Summary

• Bayes rule lets us do diagnostic queries with causal probabilities
• The naive Bayes assumption takes all features to be independent given the class label
• We can build classifiers out of a naive Bayes model using training data
• Smoothing estimates is important in real systems
• Classifier confidences are useful, when you can get them