Naïve Bayes for Text Classification

adapted by Lyle Ungar from slides by Mitch Marcus, which were adapted from slides by Massimo Poesio which were adapted from slides by Chris Manning :)}
Maximum Likelihood Estimate (MLE)

- **MLE**: The estimate that is most likely to have generated the observed data.
  - E.g., if I see 6 heads and 4 tails, what is the MLE of $P(\text{coin=heads})$?

- **Why is MLE good?**
- **Why is MLE bad?**
Example: Is this spam?

From: """" <takworlld@hotmail.com>
Subject: real estate is the only way... gem oalvgkay

Anyone can buy real estate with no money down

Stop paying rent TODAY!

There is no need to spend hundreds or even thousands for similar courses

I am 22 years old and I have already purchased 6 properties using the methods outlined in this truly INCREDIBLE ebook.

Change your life NOW!

=================================================================
Click Below to order:
http://www.wholesaledaily.com/sales/nmd.htm
=================================================================

How do you know?
Classification

◆ Given:
  ● A description of an instance, \( x \in X \), where \( X \) is the instance language or instance space.
    ■ Issue: how to represent text documents.
  ● A fixed set of categories:
    \[ C = \{ c_1, c_2, \ldots, c_k \} \]

◆ Determine:
  ● The category of \( x \): \( c(x) \in C \), where \( c(x) \) is a categorization function whose domain is \( X \) and whose range is \( C \).
    ■ We want to know how to build categorization functions ("classifiers").
A Graphical View of Text Classification

- Graphics
- Arch.
- NLP
- AI
- Theory
Examples of text categorization

- **SPAM**
  - “spam” / “not spam”

- **TOPICS**
  - “finance” / “sports” / “asia”

- **AUTHOR**
  - “Shakespeare” / “Marlowe” / “Ben Jonson”
  - The Federalist papers

- **OPINION**
  - “like” / “hate” / “neutral”
Bayesian Methods

- Uses Bayes theorem to build a generative model that approximates how data is produced.
- Uses prior probability of each category given no information about an item.
- Categorization produces a posterior probability distribution over the possible categories given a description of an item.
Bayes’ Rule once more

\[ P(C, D) = P(C \mid D)P(D) = P(D \mid C)P(C) \]

\[ P(C \mid D) = \frac{P(D \mid C)P(C)}{P(D)} \]
Maximum a posteriori (MAP)

\[ c_{MAP} \equiv \arg\max_{c \in C} P(c \mid D) \]

\[ = \arg\max_{c \in C} \frac{P(D \mid c)P(c)}{P(D)} \]

\[ = \arg\max_{c \in C} P(D \mid c)P(c) \]

As \( P(D) \) is constant
Maximum likelihood

If all hypotheses are a priori equally likely, we only need to consider the \( P(D|c) \) term:

\[
c_{ML} \equiv \arg\max_{c \in C} P(D | c)
\]

Maximum Likelihood Estimate ("MLE")
Naive Bayes Classifiers

Task: Classify a new instance $D$ based on a tuple of attribute values $D = \langle x_1, x_2, \ldots, x_n \rangle$ into one of the classes $c_j \in C$

$$c_{MAP} = \arg\max_{c \in C} P(c | x_1, x_2, \ldots, x_n)$$

$$= \arg\max_{c \in C} \frac{P(x_1, x_2, \ldots, x_n | c)P(c)}{P(x_1, x_2, \ldots, x_n)}$$

$$= \arg\max_{c \in C} P(x_1, x_2, \ldots, x_n | c)P(c)$$

Sorry: $n$ here is what we call $p$ – the number of predictors. For now we’re thinking of it as a sequence of $n$ words in a document.
Naïve Bayes Classifier: Assumption

- **$P(c_j)$**
  - Can be estimated from the frequency of classes in the training examples.

- **$P(x_1, x_2, \ldots, x_n|c_j)$**
  - $O(|X|^n \cdot |C|)$ parameters
  - Could only be estimated if a very, very large number of training examples was available.

**Naïve Bayes assumes Conditional Independence:**

- Assume that the probability of observing the conjunction of attributes is equal to the product of the individual probabilities $P(x_i|c_j)$. 
The Naïve Bayes Classifier

- **Conditional Independence Assumption:** features are independent of each other given the class:

\[ P(X_1, \ldots, X_5 \mid C) = P(X_1 \mid C) \cdot P(X_2 \mid C) \cdot \ldots \cdot P(X_5 \mid C) \]

- **This model is appropriate for binary variables**
  - Similar models work more generally
Learning the Model

◆ First attempt: maximum likelihood estimates
  • simply use the frequencies in the data

\[
\hat{P}(c_j) = \frac{N(C = c_j)}{N}
\]

\[
\hat{P}(x_i \mid c_j) = \frac{N(X_i = x_i, C = c_j)}{N(C = c_j)}
\]
Problem with Max Likelihood

What if we have seen no training cases where patient had no flu and muscle aches?

\[ P(X_1, \ldots, X_5 \mid C) = P(X_1 \mid C) \cdot P(X_2 \mid C) \cdot \cdots \cdot P(X_5 \mid C) \]

\[ \hat{P}(X_5 = t \mid C = \text{flu}) = \frac{N(X_5 = t, C = \text{flu})}{N(C = \text{flu})} = 0 \]

Zero probabilities cannot be conditioned away, no matter the other evidence!

\[ \ell = \arg \max_c \hat{P}(c) \prod_i \hat{P}(x_i \mid c) \]
Smoothing to Avoid Overfitting

\[ \hat{P}(x_i \mid c_j) = \frac{N(X_i = x_i, C = c_j) + 1}{N(C = c_j) + k} \]

◆ Somewhat more subtle version

\[ \hat{P}(x_{i,k} \mid c_j) = \frac{N(X_i = x_{i,k}, C = c_j) + mp_{i,k}}{N(C = c_j) + m} \]

- \( N(C=c_j) \) = # of docs in class \( c_j \)
- \( N(X_i=x_i,C=c_j) \) = # of docs in class \( c_j \) with word position \( X_i \) having value word \( x_i \), here \( k \) would be the vocabulary size
- If \( X_i \) is just true or false, then \( k \) is 2.
- \( p_{i,k} \) is marginalized over all classes, how often feature \( X_i \) takes on each of its \( k \) possible values.
Using Naive Bayes Classifiers to Classify Text: Bag of Words

- General model: Features are positions in the text ($X_1$ is first word, $X_2$ is second word, ...), values are words in the vocabulary

$$c_{NB} = \arg\max_{c_j \in C} P(c_j) \prod_i P(x_i \mid c_j)$$

$$= \arg\max_{c_j \in C} P(c_j) P(x_1 = \text{"our"} \mid c_j) \cdots P(x_n = \text{"text"} \mid c_j)$$

- Too many possibilities, so assume that classification is independent of the positions of the words
  - Result is bag of words model
  - Just use the counts of words, or even a variable for each word: is it in the document or not?
Smoothing to Avoid Overfitting – Bag of words

\[
\hat{P}(x_i \mid c_j) = \frac{N(X_i = true, C = c_j)}{N(C = c_j)} + \frac{1}{k} + \frac{m p_i}{N(C = c_j) + m}
\]

- Somewhat more subtle version

Now

\(N(C=c_j)\) = # of docs in class \(c_j\)
\(N(X_i=true, C=c_j)\) = # of docs in class \(c_j\) containing word \(x_i\)
\(k\) = vocabulary size
\(p_i\) is the the probability that word \(i\) is present, ignoring class labels

# of values of \(X_i\)

overall fraction of docs containing \(x_i\)

extent of “smoothing”
Naïve Bayes: Learning

◆ From training corpus, determine Vocabulary

◆ Estimate $P(c_j)$ and $P(x_k \mid c_j)$
  
  - For each $c_j$ in C do
    
    $\text{docs}_j \leftarrow$ documents labeled with class $c_j$

    $$P(c_j) \leftarrow \frac{|\text{docs}_j|}{|\text{total \# documents}|}$$

  
  - For each word $x_k$ in Vocabulary
    
    $n_k \leftarrow$ number of occurrences of $x_k$ in all $\text{docs}_j$

    $$P(x_k \mid c_j) \leftarrow \frac{n_k + 1}{|\text{docs}_j| + |\text{Vocabulary}|}$$  \text{Simple “Laplace” smoothing}
Naïve Bayes: Classifying

- For all words $i$ in current document
- Return $c_{NB}$, where

$$c_{NB} = \arg\max_{c_j \in C} P(c_j) \prod_{i \in \text{document}} P(x_i | c_j)$$

What is the implicit assumption hidden in this?
Naïve Bayes for text

- The “correct” model would have a probability for each word observed and one for each word \textit{not} observed.
  - Naïve Bayes for text assumes that there is no information in words that are not observed – since most words are very rare, their probability of \textit{not} being seen is close to 1.
Naive Bayes is not so dumb

- A good dependable baseline for text classification
  - But not the best!

- Optimal if the Independence Assumptions hold:
  - If assumed independence is correct, then it is the Bayes Optimal Classifier for problem

- Very Fast:
  - Learn with one pass over the data;
  - Testing linear in the number of attributes, and document collection size

- Low Storage requirements
Technical Detail: Underflow

- Multiplying lots of probabilities, which are between 0 and 1 by definition, can result in floating-point underflow.

- Since $\log(xy) = \log(x) + \log(y)$, it is better to perform all computations by summing logs of probabilities rather than multiplying probabilities.

- Class with highest final un-normalized log probability score is still the most probable.

\[
c_{NB} = \arg\max_{c_j \in C} \log P(c_j) + \sum_{i \in \text{positions}} \log P(x_i | c_j)
\]
More Facts About Bayes Classifiers

- Bayes Classifiers can be built with real-valued inputs*
  - Or many other distributions
- Bayes Classifiers don’t try to be maximally discriminative
  - They merely try to honestly model what’s going on*
- Zero probabilities give stupid results
- Naïve Bayes is wonderfully cheap
  - And handles 1,000,000 features cheerfully!

*See future Lectures
Naïve Bayes – MLE

<table>
<thead>
<tr>
<th>word</th>
<th>topic</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>sports</td>
<td>0</td>
</tr>
<tr>
<td>ball</td>
<td>sports</td>
<td>1</td>
</tr>
<tr>
<td>carrot</td>
<td>sports</td>
<td>0</td>
</tr>
<tr>
<td>game</td>
<td>sports</td>
<td>2</td>
</tr>
<tr>
<td>I</td>
<td>sports</td>
<td>2</td>
</tr>
<tr>
<td>saw</td>
<td>sports</td>
<td>2</td>
</tr>
<tr>
<td>the</td>
<td>sports</td>
<td>3</td>
</tr>
</tbody>
</table>

**Assume 5 sports documents**

Counts are number of documents on the sports topic containing each word

\[
P(a|\text{sports}) = \frac{0}{5} \\
P(\text{ball}|\text{sports}) = \frac{1}{5}
\]
### Naïve Bayes – prior (noninformative)

<table>
<thead>
<tr>
<th>Word</th>
<th>topic</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>sports</td>
<td>0.5</td>
</tr>
<tr>
<td>ball</td>
<td>sports</td>
<td>0.5</td>
</tr>
<tr>
<td>carrot</td>
<td>sports</td>
<td>0.5</td>
</tr>
<tr>
<td>game</td>
<td>sports</td>
<td>0.5</td>
</tr>
<tr>
<td>l</td>
<td>sports</td>
<td>0.5</td>
</tr>
<tr>
<td>saw</td>
<td>sports</td>
<td>0.5</td>
</tr>
<tr>
<td>the</td>
<td>sports</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Assume 5 sports documents

Adding a count of 1 (instead of 0.5 as done here) is called “Laplace smoothing”.

Pseudo-counts to be added to the observed counts

We did 0.5 here; before in the notes it was 1; either is fine
Naïve Bayes – posterior (MAP)

<table>
<thead>
<tr>
<th>Word</th>
<th>topic</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>sports</td>
<td>0.5</td>
</tr>
<tr>
<td>ball</td>
<td>sports</td>
<td>1.5</td>
</tr>
<tr>
<td>carrot</td>
<td>sports</td>
<td>0.5</td>
</tr>
<tr>
<td>game</td>
<td>sports</td>
<td>2.5</td>
</tr>
<tr>
<td>I</td>
<td>sports</td>
<td>2.5</td>
</tr>
<tr>
<td>saw</td>
<td>sports</td>
<td>2.5</td>
</tr>
<tr>
<td>the</td>
<td>sports</td>
<td>3.5</td>
</tr>
</tbody>
</table>

Assume 5 sports documents,

\[
P(\text{word,topic}) = \frac{N(\text{word,topic})+0.5}{N(\text{topic}) + 0.5 \times k}
\]

Pseudo count of docs on topic=sports is \((5 + 0.5\times7=8.5)\)

\[
P(a|\text{sports}) = \frac{0.5}{8.5} \quad \text{posterior}
\]

\[
P(\text{ball}|\text{sports}) = \frac{1.5}{8.5}
\]
# Naïve Bayes – prior overall

<table>
<thead>
<tr>
<th>word</th>
<th>topic</th>
<th>count</th>
<th>topic</th>
<th>count</th>
<th>p(word)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>sports</td>
<td>0</td>
<td>politics</td>
<td>2</td>
<td>2/11</td>
</tr>
<tr>
<td>ball</td>
<td>sports</td>
<td>1</td>
<td>politics</td>
<td>0</td>
<td>1/11</td>
</tr>
<tr>
<td>carrot</td>
<td>sports</td>
<td>0</td>
<td>politics</td>
<td>0</td>
<td>0/11</td>
</tr>
<tr>
<td>game</td>
<td>sports</td>
<td>2</td>
<td>politics</td>
<td>1</td>
<td>3/11</td>
</tr>
<tr>
<td>l</td>
<td>sports</td>
<td>2</td>
<td>politics</td>
<td>5</td>
<td>7/11</td>
</tr>
<tr>
<td>saw</td>
<td>sports</td>
<td>2</td>
<td>politics</td>
<td>1</td>
<td>3/11</td>
</tr>
<tr>
<td>the</td>
<td>sports</td>
<td>3</td>
<td>politics</td>
<td>5</td>
<td>8/11</td>
</tr>
</tbody>
</table>

Assume 5 sports docs and 6 politics docs 11 total docs
Naïve Bayes – posterior (MAP)

\[ P(a|\text{sports}) = \frac{0 + 4 \times (2/11)}{5 + 4} = 0.08 \]
\[ P(\text{ball}|\text{sports}) = \frac{1 + 4 \times (1/11)}{5 + 4} = 0.15 \]

\[ P(\text{word,topic}) = \frac{N(\text{word,topic}) + 4}{N(\text{topic}) + 4} P_{\text{word}} \]

Here we arbitrarily pick \( m=4 \) as the strength of our prior.
What you should know

Applications of document classification

- Spam detection, topic prediction, email routing, author ID, sentiment analysis

Naïve Bayes

- As MAP estimator (uses prior)
  - Contrast MLE
  - Smoothing
- For document classification
  - Use bag of words
  - Could use richer feature set
Regression Search

Given $p$ features of which $q$ end up being selected:

- Stepwise regression will estimate ...
  
  A) $q$ regressions
  B) $q \cdot p$ regressions
  C) $p^2$ regressions
  D) $q \cdot p^2$ regressions

- Streamwise regression will estimate ...

The largest matrix to be inverted by stepwise regression is

A) $1 \times 1$   B) $q \times q$   C) $p \times p$

The largest matrix to be inverted by stagewise regression is

...
Regression Search

◆ Stepwise regression is used to minimize

A) Training set error (MLE)
B) $L_0$ penalized training set error
C) any penalized training set error