Naïve Bayes for Text Classification

(adapted from slides by Mitch Marcus, which were adapted from slides by Massimo Poesio which were adapted from slides by Chris Manning)

MLE

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:
    $C = \{c_1, c_2, \ldots, c_n\}$
- 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

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

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

Maximum a posteriori (MAP)

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

As \( P(D) \) is constant

Maximum likelihood

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

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

Naive Bayes Classifiers

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

\[ c_{MAP} = \arg \max_{c \in C} P(c \mid x_1, x_2, \ldots, x_n) = \arg \max_{c \in C} \frac{P(x_1, x_2, \ldots, x_n \mid c)P(c)}{P(x_1, x_2, \ldots, x_n)} = \arg \max_{c \in C} P(x_1, x_2, \ldots, x_n \mid c)P(c) \]

Naïve Bayes Classifier: Assumption

- \( P(c) \)
  - Can be estimated from the frequency of classes in the training examples.
- \( P(x_1, x_2, \ldots, x_n \mid c) \)
  - \( O(|X|^{|C|}) \) parameters
  - Could only be estimated if a very, very large number of training examples was available.

Naive Bayes Conditional Independence Assumption:

- Assume that the probability of observing the conjunction of attributes is equal to the product of the individual probabilities \( P(x_j \mid c) \).
The Naïve Bayes Classifier

- **Conditional Independence Assumption**: features are independent of each other given the class:
  \[ P(X_1, \ldots, X_n | C) = P(X_1 | C) \cdot P(X_2 | C) \cdot \ldots \cdot P(X_n | C) \]

- This model is appropriate for binary variables

Problem with Max Likelihood

- What if we have seen no training cases where patient had no flu and muscle aches?
  \[ P(X_1, X_2 | C) = P(X_1 | C) \cdot P(X_2 | C) \cdot \ldots \cdot P(X_n | C) \]
  \[ \hat{P}(X_1 = t | C = \text{flu}) = \frac{N(X_1 = t, C = \text{flu})}{N(C = \text{flu})} = 0 \]

- Zero probabilities cannot be conditioned away, no matter the other evidence!
  \[ \ell = \arg \max C \hat{P}(C) \prod x \hat{P}(x | C) \]

Using Naïve Bayes Classifiers to Classify Text: Basic method

- Attributes are text positions, values are words.
  \[ c_{ik} = \arg \max_c P(c) \prod_i P(x_i | c) \]
  \[ = \arg \max_c P(c_1) P(x_{1\text{text}} | c) \cdots P(x_{n\text{text}} | c) \]

- Still too many possibilities
- Assume that classification is independent of the positions of the words
  - Use same parameters for each position
  - Result is bag of words model

Learning the Model

- **First attempt**: maximum likelihood estimates
  - simply use the frequencies in the data
    \[ \hat{P}(c_j) = \frac{N(C = c_j)}{N} \]
    \[ \hat{P}(x_i | c_j) = \frac{N(X_i = x_i, C = c_j)}{N(C = c_j)} \]

Smoothing to Avoid Overfitting

- Somewhat more subtle version
  \[ \hat{P}(x_{ik} | c_j) = \frac{N(X_i = x_{ik}, C = c_j) + m}{N(C = c_j) + N} \]
  \[ m \text{ of values of } X_k \]

Naïve Bayes: Learning

- From training corpus, extract **Vocabulary**
- Calculate required \( P(c) \) and \( P(x_i | c_j) \) terms
  - For each \( c_j \) in \( C \) do
    \[ \text{docs}_{ij} \leftarrow \text{subset of documents for which the target class is } c_j \]
    \[ P(c_j) \leftarrow \frac{| \text{docs}_{ij} |}{| \text{total docs} |} \]
  - For each word \( x_k \) in **Vocabulary**
    \[ n_k \leftarrow \text{number of occurrences of } x_k \text{ in all docs}_{ij} \]
    \[ P(x_k | c_j) = \frac{n_k + 1}{| \text{docs}_{ij} | + | \text{Vocabulary} |} \]
Naïve Bayes: Classifying

- positions ← all word positions in current document which contain tokens found in Vocabulary
- Return $c_{NB}$ where
  $$c_{NB} = \arg\max_{c \in C} P(c) \prod_{i \in \text{positions}} P(x_i \mid c)$$

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 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 not being seen is close to 1.

Pantel and Lin: SPAMCOP

- Uses a Naïve Bayes classifier
- M is spam if $P(\text{Spam}|M) / P(\text{NonSpam}|M) > \text{threshold}$
- Method
  - Tokenize message and stem using Porter Stemmer
  - Estimate $P(W|C)$ using m-estimate (a form of smoothing)
  - Remove words that do not satisfy certain conditions
  - Train: 160 spams, 466 non-spams
  - Test: 277 spams, 346 non-spams
- Results: ERROR RATE of 4.33%
  - Worse results using trigrams

Naive Bayes is Not So Naive

- 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:
  - Learning with one pass over the data;
  - Testing linear in the number of attributes, and document collection size
- Low Storage requirements

Technical Detail: Underflow Prevention

- 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.
  $$c_{NB} = \arg\max_{c \in C} \log P(c) + \sum_{i \in \text{positions}} \log P(x_i \mid c)$$

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 survives 1,000,000 features cheerfully!

*See future Lectures
Naïve Bayes – MLE vs 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</td>
</tr>
<tr>
<td>baseball</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>
<tr>
<td>a</td>
<td>politics</td>
<td>2</td>
</tr>
</tbody>
</table>

\[ P(\text{game}|\text{sports}) = \frac{2}{10} \quad p(\text{a}|\text{sports}) = \frac{0}{10} \]

Naïve Bayes – prior

<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>baseball</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>I</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>
<tr>
<td>a</td>
<td>politics</td>
<td>0.5</td>
</tr>
</tbody>
</table>

\[ P(\text{game}|\text{sports}) = \frac{1}{7} \quad p(\text{a}|\text{sports}) = \frac{1}{7} \]

Naïve Bayes – posterior

<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>baseball</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>
<tr>
<td>a</td>
<td>politics</td>
<td>2.5</td>
</tr>
</tbody>
</table>

\[ P(\text{game}|\text{sports}) = \frac{2.5}{13.5} \quad p(\text{a}|\text{sports}) = \frac{0.5}{13.5} \]

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