Naïve Bayes

- A special type of Bayesian network
- Makes a conditional independence assumption
- Typically used for classification

Classification

Suppose you are trying to classify situations that determine whether or not Blackboard will be down. You’ve come up with the following list of variables (which are all Boolean):

<table>
<thead>
<tr>
<th>Monday</th>
<th>Assn</th>
<th>Grades</th>
<th>Win</th>
<th>BD</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>false</td>
<td>false</td>
<td>true</td>
</tr>
</tbody>
</table>

We also have a Boolean variable called BD which stands for “Blackboard down”

These are called features or attributes
This is called the “class” variable (because we’re trying to classify it)

Notice the conditional independence assumption:
The features are conditionally independent given the class variable.
Naïve Bayes Parameters

\( P(BD) = ? \)

\( P(M | BD) = ? \)  \( P(A | BD) = ? \)  \( P(G | BD) = ? \)  \( P(W | BD) = ? \)

How do you get these parameters from the training data?

Naïve Bayes Parameters

\( BD \)

\( M \) \( A \) \( G \) \( W \)

\( P(BD) \)

<table>
<thead>
<tr>
<th>BD</th>
<th>( P(BD) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>( \frac{\text{(# of records in training data with } BD = \text{false})}{\text{(# of records in training data)}} )</td>
</tr>
<tr>
<td>true</td>
<td>( \frac{\text{(# of records in training data with } BD = \text{true})}{\text{(# of records in training data)}} )</td>
</tr>
</tbody>
</table>

Inference in Naïve Bayes

\( P(BD | M, A, G, W) \)

By Bayes Rule

\[= \frac{P(M, A, G, W | BD)P(BD)}{P(M, A, G, W)}\]

\[= aP(M, A, G, W | BD)P(BD)\]

Treat denominator as constant

\[= aP(BD)P(M | BD)P(A | BD)P(G | BD)P(W | BD)\]

From conditional independence

Prediction

- Suppose you are now in a day when \( M=\text{true}, A=\text{true}, G=\text{true}, W=\text{true} \).
- You need to predict if \( BD=\text{true} \) or \( BD=\text{false} \).
- We will use the notation that \( BD=\text{true} \) is equivalent to \( bd \) and \( BD=\text{false} \) is equivalent to \( \neg bd \).
The General Case

1. Estimate $P(Y=v)$ as fraction of records with $Y=v$
2. Estimate $P(X_i=u | Y=v)$ as fraction of “$Y=v$” records that also have $X=u$.
3. To predict the $Y$ value given observations of all the $X_i$ values, compute

$$Y_{\text{predict}} = \arg\max_v P(Y = v | X_1 = u_1, \ldots, X_n = u_n)$$

Naive Bayes Classifier

$$Y_{\text{predict}} = \arg\max_v P(Y = v | X_1 = u_1, \ldots, X_n = u_n)$$

Technical Point #1

- The probabilities $P(X_i = u_j | Y = v)$ can sometimes be really small
- This can result in numerical instability since floating point numbers are not represented exactly on any computer architecture
- To get around this, use the log of the last line in the previous slide ie.

$$Y_{\text{predict}} = \arg\max_v \left[ \log(P(Y = v)) + \sum_{j=1}^n \log(P(X_j = u_j | Y = v)) \right]$$

Technical Point #2

- When estimating parameters, what happens if you don’t have any records that match a certain combination of features?
- For example, in our training data, we didn’t have $M=false, A=false, G=false, W=false$
- This means that $P(X_i = u_j | Y = v)$ in the formula below will be 0 and the entire expression will be 0.

$$P(Y = v) \prod_{j=1}^n P(X_j = u_j | Y = v)$$

Dirichlet Priors

Let $N_j$ be the number of values that $X_j$ can take on.

$$P(X_j = u_j | V = v) = \frac{(# \text{records with } X_j = u_j \text{ and } V = v) + 1}{(# \text{records with } V = v) + N_j}$$

What happens when you have no records with $V = v$?

$$P(X_j = u_j | V = v) = \frac{1}{N_j}$$

This means that each value of $X_j$ is equally likely in the absence of data. If you have a lot of data, it dominates the $1/N_j$ value. We call this trick a “Dirichlet Prior”.

What You Should Know

- How to learn the parameters for a Naïve Bayes model
- How to make predictions with a Naïve Bayes model
- How to implement a Naïve Bayes Model
Programming Assignment #3

You will classify text into 2 classes
Specifically, you will classify fortune cookie messages into 2 groups: fortune and non-fortune

There are three sets of files:
1. Training data: traintdata.txt
2. Testing data: testdata.txt
3. Vocabulary: vocabulary.txt

Two parts to this assignment:
1. Pre-processing step
2. Classification step

The Vocabulary
• The features of the documents will be the presence/absence of words in the vocabulary
• The vocabulary is the list of words that are known to the classifier
• Ideally, the vocabulary would be all the words in the English language
• For this assignment, we will set the vocabulary to be the union of the words in the training data (almost... see the next few slides)

Bag of Words
Suppose you have the following documents in the training data:

<table>
<thead>
<tr>
<th>Class Label</th>
<th>Class 1</th>
<th>Class 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>You will have a pleasant experience</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adventure can be real happiness</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The vocabulary will be:
you, will, have, a, pleasant, experience, adventure, can, be, real, happiness
Bag of Words

Vocab: you, will, a, pleasant, experience, adventure, can, be, real, happiness

- Not all of these words are useful for classification. For example, the ones in red above are kind of boring and uninformative.
- We call these stop words.
- Typically, the stop words are provided in a file.
- The stop words are removed from the vocabulary

Vocab: will, pleasant, experience, adventure, real, happiness

Keep this in alphabetical order to help with debugging

Vocab: adventure, experience, happiness, pleasant, real, will

Training data

- Now that you have your vocabulary, you need to convert your training data into features

<table>
<thead>
<tr>
<th>Class Label</th>
<th>Adventure can be real happiness</th>
<th>Will have a pleasant experience</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class 0</td>
<td>0 0 0 0 1 0</td>
<td>1 1 1 0 1 0</td>
</tr>
<tr>
<td>Class 1</td>
<td>1 0 0 0 0 0</td>
<td>0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

You will output the training data in feature form, with the features alphabetized (we will mark you on this output).

2. Classification Step (Training Phase)

- Your naive Bayes classifier now looks something like the above
- You still need to fill in the conditional probability tables in each node
- This is done in the training phase (as described on slides 9 and 10)
- Remember to use the dirichlet prior trick (see slide 17)

2. Classification Step (Testing Phase)

- Testing phase: you now need to load the test data and convert it into feature form
- ie. express the document as the presence/absence of words in the vocabulary

<table>
<thead>
<tr>
<th>Class Label</th>
<th>Adventure can be real happiness</th>
<th>Will have a pleasant weekend</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class 0</td>
<td>0 0 0 0 0 0</td>
<td>1 0 0 1 1 0</td>
</tr>
<tr>
<td>Class 1</td>
<td>1 0 0 0 0 0</td>
<td>0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

Note:
- Ignore words not in your vocabulary (eg. weekend)
- The testing data set in feature form does not have a class label column

Results

There are two sets of results for the baseline classifier:

1. Results #1:
   - Use traindata.txt for the training phase
   - Use traindata.txt for the testing phase
   - Report accuracy

2. Results #2:
   - Use traindata.txt for the training phase
   - Use testdata.txt for the testing phase
   - Report accuracy