SI485i : NLP

Set 5
Using Naïve Bayes
Motivation

• We want to predict *something*.
• We have some *text* related to this *something*.

• *something* = target label \( Y \)
• *text* = text features \( X \)

*Given \( X \), what is the most probable \( Y \)?*
Motivation: Author Detection

$X = \text{Alas the day! take heed of him; he stabbed me in mine own house, and that most beastly: in good faith, he cares not what mischief he does. If his weapon be out: he will foin like any devil; he will spare neither man, woman, nor child.}$

$Y = \{ \text{Charles Dickens, William Shakespeare, Herman Melville, Jane Austin, Homer, Leo Tolstoy} \}$

$$Y \leftarrow \arg \max_{y_k} P(Y = y_k) P(X \mid Y = y_k)$$
More Motivation

\[ P(Y=\text{spam} \mid X=\text{email}) \]

\[ P(Y=\text{worthy} \mid X=\text{review sentence}) \]
The Naïve Bayes Classifier

- Recall Bayes rule: 
  \[ P(Y_i \mid X_j) = \frac{P(Y_i)P(X_j \mid Y_i)}{P(X_j)} \]

- Which is short for:
  \[ P(Y = y_i \mid X = x_j) = \frac{P(Y = y_i)P(X = x_j \mid Y = y_i)}{P(X = x_j)} \]

- We can re-write this as:
  \[ P(Y = y_i \mid X = x_j) = \frac{P(Y = y_i)P(X = x_j \mid Y = y_i)}{\sum_k P(Y = y_k)P(X = x_j \mid Y = y_k)} \]

Remaining slides adapted from Tom Mitchell.
Deriving Naïve Bayes

• Idea: use the training data to directly estimate:

\[
P(X \mid Y) \quad \text{and} \quad P(Y)
\]

• We can use these values to estimate using Bayes rule.

\[
P(Y \mid X)
\]

• Recall that representing the full joint probability is not practical.

\[
P(X \mid Y) = P(X_1, X_2, \ldots, X_n \mid Y)
\]
Deriving Naïve Bayes

• However, if we make the assumption that the attributes are independent, estimation is easy!

\[ P(X_1, \ldots, X_n \mid Y) = \prod_{i} P(X_i \mid Y) \]

• In other words, we assume all attributes are conditionally independent given \( Y \).
  • Often this assumption is violated in practice, but more on that later…
Deriving Naïve Bayes

Let \( X = \langle X_1, \ldots, X_n \rangle \) and label \( Y \) be discrete.

Then, we can estimate \( P(X_i \mid Y_i) \) and \( P(Y_i) \) directly from the training data by counting!

<table>
<thead>
<tr>
<th>Sky</th>
<th>Temp</th>
<th>Humid</th>
<th>Wind</th>
<th>Water</th>
<th>Forecast</th>
<th>Play?</th>
</tr>
</thead>
<tbody>
<tr>
<td>sunny</td>
<td>warm</td>
<td>normal</td>
<td>strong</td>
<td>warm</td>
<td>same</td>
<td>yes</td>
</tr>
<tr>
<td>sunny</td>
<td>warm</td>
<td>high</td>
<td>strong</td>
<td>warm</td>
<td>same</td>
<td>yes</td>
</tr>
<tr>
<td>rainy</td>
<td>cold</td>
<td>high</td>
<td>strong</td>
<td>warm</td>
<td>change</td>
<td>no</td>
</tr>
<tr>
<td>sunny</td>
<td>warm</td>
<td>high</td>
<td>strong</td>
<td>cool</td>
<td>change</td>
<td>yes</td>
</tr>
</tbody>
</table>

\[ P(\text{Sky} = \text{sunny} \mid \text{Play} = \text{yes}) = ? \quad P(\text{Humid} = \text{high} \mid \text{Play} = \text{yes}) = ? \]
The Naïve Bayes Classifier

Now we have:

\[
P(Y = y_j \mid X_1, \ldots, X_n) = \frac{P(Y = y_j) \prod_i P(X_i \mid Y = y_j)}{\sum_k P(Y = y_k) \prod_i P(X_i \mid Y = y_k)}
\]

To classify a new point \( X_{\text{new}} \):

\[
Y_{\text{new}} \leftarrow \arg \max_{y_k} P(Y = y_k) \prod_i P(X_i \mid Y = y_k)
\]
Represent LMs as NB prediction

$X =$ your text!
$P(X) :$ your language model does this!
$P(X \mid Y) :$ still a language model, but trained on Y data

<table>
<thead>
<tr>
<th>Y1 = dickens</th>
<th>Y2 = twain</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(Y1) \times P(X \mid Y1)$</td>
<td>$P(Y2) \times P(X \mid Y2)$</td>
</tr>
<tr>
<td>$P(X \mid Y1) = P_{Y1}(X)$</td>
<td>$P(X \mid Y2) = P_{Y2}(X)$</td>
</tr>
<tr>
<td>Bigrams: $P_{Y1}(X) = \prod_i P_{Y1}(x_i \mid x_i - 1)$</td>
<td>Bigrams: $P_{Y2}(X) = \prod_i P_{Y2}(x_i \mid x_i - 1)$</td>
</tr>
</tbody>
</table>

$X =$ your text!
$P(X) :$ your language model does this!
$P(X \mid Y) :$ still a language model, but trained on Y data
Naïve Bayes Applications

- Text classification
  - Which e-mails are spam?
  - Which e-mails are meeting notices?
  - Which author wrote a document?
  - Which webpages are about current events?
  - Which blog contains angry writing?
  - What sentence in a document talks about company X?
  - etc.
Text and Features

\[ P(X_1, \ldots, X_n \mid Y) = \prod_i P(X_i \mid Y) \]

• What is \(X_i\)?
  • Could be unigrams, hopefully bigrams too.

• It can be *anything* that is computed from the text \(X\).
  • Yes, I really mean anything. Creativity and intuition into language is where the real gains come from in NLP.

• Non n-gram examples:
  • \(X_{10} = \text{“the sentence contains a conjunction (yes or no)”} \)
  • \(X_{356} = \text{“existence of a semi-colon in the paragraph”} \)
Features

• In machine learning, “features” are the attributes to which you assign weights (probabilities in Naïve Bayes) that help in the final classification.

• Up until now, your features have been n-grams. You now want to consider other types of features.
  • You count features just like n-grams. How many did you see?

• \( X = \) set of features
• \( P(Y|X) = \) probability of a \( Y \) given a set of features
How do you count features?

• Feature idea: “a semicolon exists in this sentence”

• Count them:
  • Count("FEAT-SEMICOLON", 1)
  • *Make up a unique name for the feature, then count!*

• Compute probability:
  • $P(\text{"FEAT-SEMICOLON" | author="dickens"}) = \frac{\text{Count("FEAT-SEMICOLON")}}{\text{(# dickens sentences)}}$
Authorship Lab

1. Figure out how to use your Language Models from Lab 2. They can be your initial features.
   • Can you train() a model on one author’s text?

2. \( P(\text{dickens} \mid \text{text}) = P(\text{dickens}) \times P_{\text{BigramModel}}(\text{text}) \)

3. New code for new features. Call your language models, get a probability, and then multiply new feature probabilities.