text classification 1: naive Bayes

CS 585, Fall 2018
Introduction to Natural Language Processing
http://people.cs.umass.edu/~miyyer/cs585/

Mohit Iyyer
College of Information and Computer Sciences
University of Massachusetts Amherst
questions after last class...

• can you post LaTeX source of the HW?
• in HW0 q4, what is meant by dimensionality?
• what is the course load like? how often will you be giving out HWs and what will be the usual split of theoretical / coding questions?
text classification

- input: some text \( x \) (e.g., sentence, document)
- output: a label \( y \) (from a finite label set)
- goal: learn a mapping function \( f \) from \( x \) to \( y \)
<table>
<thead>
<tr>
<th>problem</th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>sentiment analysis</td>
<td>text from reviews (e.g., IMDB)</td>
<td>{positive, negative}</td>
</tr>
<tr>
<td>topic identification</td>
<td>documents</td>
<td>{sports, news, health, …}</td>
</tr>
<tr>
<td>author identification</td>
<td>books</td>
<td>{Tolkien, Shakespeare, …}</td>
</tr>
<tr>
<td>spam identification</td>
<td>emails</td>
<td>{spam, not spam}</td>
</tr>
</tbody>
</table>

... many more!
input $\mathbf{x}$:

```
From European Union <info@eu.org>
Subject
Reply to

Please confirm to us that you are the owner of this very email address with your copy of identity card as proof.

YOU EMAIL ID HAS WON $10,000,000.00 ON THE ONGOING EUROPEAN UNION COMPENSATION FOR SCAM VICTIMS. CONTACT OUR EMAIL: [REDACTED] NOW TO CLAIM YOUR COMPENSATION
```

label space: spam or not spam

we’d like to learn a mapping $f$ such that $f(\mathbf{x}) = \text{spam}$
$f$ can be hand-designed rules

- if “won $10,000,000” in $x$, $y = \text{spam}$
- if “CS585 Fall 2018” in $x$, $y = \text{not spam}$

what are the drawbacks of this method?
$f$ can be learned from data

- given **training data** (already-labeled $x,y$ pairs) learn $f$ by maximizing the likelihood of the training data

- this is known as **supervised learning**
probability review

• random variable $X$ takes value $x$ with probability $p(X = x)$; shorthand $p(x)$

• joint probability: $p(X = x, Y = y)$

• conditional probability: $p(X = x \mid Y = y)$

$$p(X = x \mid Y = y) = \frac{p(X = x, Y = y)}{p(Y = y)}$$

• when does $p(X = x, Y = y) = p(X = x) \cdot p(Y = y)$?
probability of some input text

- goal: assign a probability to a sentence
  - sentence: sequence of tokens
    \[ p(w_1, w_2, w_3, \ldots, w_n) \]
    \[ p(\text{the cat sleeps}) > p(\text{cat sleeps the}) \]
  - \( w_i \in V \) where \( V \) is the vocabulary (types)
- some constraints:
  - non-negativity: for any \( w \in V, \ p(w) \geq 0 \)
  - probability distribution, sums to 1
    \[ \sum_{w \in V} p(w) = 1 \]
how to estimate $p(\text{sentence})$?

$$p(w_1, w_2, w_3, \ldots, w_n)$$

we could count all occurrences of the sequence

$$w_1, w_2, w_3, \ldots, w_n$$

in some large dataset and normalize by the number of sequences of length $n$ in that dataset

how many parameters would this require?
chain rule

\[ p(w_1, w_2, w_3, \ldots, w_n) \]

\[ = p(w_1) \cdot p(w_2 | w_1) \cdot p(w_3 | w_1, w_2) \ldots \cdot p(w_n | w_1 \ldots n-1) \]

in naive Bayes, the probability of generating a word is independent of all other words

\[ = p(w_1) \cdot p(w_2) \cdot p(w_3) \ldots \cdot p(w_n) \]

this is called the unigram probability. what are its limitations?
toy sentiment example

• vocabulary V: {i, hate, love, the, movie, actor}
• reviews:
  • i hate the movie
  • i love the movie
  • i hate the actor
  • the movie i love
  • i love love love love love love the movie
  • hate movie
  • i hate the actor i love the movie
# bag-of-words representation

i hate the actor i love the movie

<table>
<thead>
<tr>
<th>word</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>2</td>
</tr>
<tr>
<td>hate</td>
<td>1</td>
</tr>
<tr>
<td>love</td>
<td>1</td>
</tr>
<tr>
<td>the</td>
<td>2</td>
</tr>
<tr>
<td>movie</td>
<td>1</td>
</tr>
<tr>
<td>actor</td>
<td>1</td>
</tr>
</tbody>
</table>

equivalent representation to:
actor i i the love the movie hate
naive Bayes

- represents input text as a bag of words
- assumption: each word is independent of all other words
- given labeled data, we can use naive Bayes to estimate probabilities for unlabeled data
- **goal**: infer probability distribution that generated the labeled data for each label
which of the below distributions most likely generated the positive reviews?
... back to our reviews

\[ p(\text{I love love love love love love the movie}) = p(\text{I}) \cdot p(\text{love})^5 \cdot p(\text{the}) \cdot p(\text{movie}) \]

\[ = 5.95374181 \times 10^{-7} \]

\[ = 1.4467592 \times 10^{-4} \]
logs to avoid underflow

\[ p(w_1) \cdot p(w_2) \cdot p(w_3) \cdots p(w_n) \]

can get really small esp. with large \( n \)

\[
\log \prod p(w_i) = \sum \log p(w_i)
\]

\[
p(i) \cdot p(\text{love})^5 \cdot p(\text{the}) \cdot p(\text{movie}) = 5.95374181e-7
\]

\[
\log p(i) + 5 \log p(\text{love}) + \log p(\text{the}) + \log p(\text{movie})
\]

\[
= -14.3340757538
\]
class conditional probabilities

Bayes rule (ex: \( x = \) sentence, \( y = \) label in \{pos, neg\})

\[
p(y \mid x) = \frac{p(y) \cdot P(x \mid y)}{p(x)}
\]

our predicted label is the one with the highest posterior probability, i.e.,

\[
\hat{y} = \arg \max_{y \in Y} p(y) \cdot P(x \mid y)
\]

derive!
Remember the independence assumption!

\[
\hat{y} = \arg \max_{y \in Y} p(y) \cdot P(x \mid y)
\]

\[
= \arg \max_{y \in Y} p(y) \cdot \prod_{w \in x} P(w \mid y)
\]

\[
= \arg \max_{y \in Y} \log p(y) + \sum_{w \in x} \log P(w \mid y)
\]
computing the prior...

- i hate the movie
- i love the movie
- i hate the actor
- the movie i love
- i love love love love love the movie
- hate movie
- i hate the actor i love the movie

$p(y)$ lets us encode inductive bias about the labels we can estimate it from the data by simply counting...

<table>
<thead>
<tr>
<th>label y</th>
<th>count</th>
<th>p(Y=y)</th>
<th>log(p(Y=y))</th>
</tr>
</thead>
<tbody>
<tr>
<td>positive</td>
<td>3</td>
<td>0.43</td>
<td>-0.84</td>
</tr>
<tr>
<td>negative</td>
<td>4</td>
<td>0.57</td>
<td>-0.56</td>
</tr>
</tbody>
</table>
computing the likelihood...

<table>
<thead>
<tr>
<th>word</th>
<th>count</th>
<th>( p(w \mid y) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>3</td>
<td>0.19</td>
</tr>
<tr>
<td>hate</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>love</td>
<td>7</td>
<td>0.44</td>
</tr>
<tr>
<td>the</td>
<td>3</td>
<td>0.19</td>
</tr>
<tr>
<td>movie</td>
<td>3</td>
<td>0.19</td>
</tr>
<tr>
<td>actor</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td><strong>total</strong></td>
<td><strong>16</strong></td>
<td></td>
</tr>
</tbody>
</table>

\[ p(X \mid y=\text{positive}) \]

<table>
<thead>
<tr>
<th>word</th>
<th>count</th>
<th>( p(w \mid y) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>4</td>
<td>0.22</td>
</tr>
<tr>
<td>hate</td>
<td>4</td>
<td>0.22</td>
</tr>
<tr>
<td>love</td>
<td>1</td>
<td>0.06</td>
</tr>
<tr>
<td>the</td>
<td>4</td>
<td>0.22</td>
</tr>
<tr>
<td>movie</td>
<td>3</td>
<td>0.17</td>
</tr>
<tr>
<td>actor</td>
<td>2</td>
<td>0.11</td>
</tr>
<tr>
<td><strong>total</strong></td>
<td><strong>18</strong></td>
<td></td>
</tr>
</tbody>
</table>

\[ p(X \mid y=\text{negative}) \]
The table shows the word frequencies and their conditional probabilities for two different sentiments. The left side is for positive sentiment, and the right side is for negative sentiment.

<table>
<thead>
<tr>
<th>word</th>
<th>count</th>
<th>( p(w \mid y) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>3</td>
<td>0.19</td>
</tr>
<tr>
<td>hate</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>love</td>
<td>7</td>
<td>0.44</td>
</tr>
<tr>
<td>the</td>
<td>3</td>
<td>0.19</td>
</tr>
<tr>
<td>movie</td>
<td>3</td>
<td>0.19</td>
</tr>
<tr>
<td>actor</td>
<td>0</td>
<td>0.00</td>
</tr>
</tbody>
</table>

**Total**: 16

For the negative sentiment, the table looks similar:

<table>
<thead>
<tr>
<th>word</th>
<th>count</th>
<th>( p(w \mid y) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>4</td>
<td>0.22</td>
</tr>
<tr>
<td>hate</td>
<td>4</td>
<td>0.22</td>
</tr>
<tr>
<td>love</td>
<td>1</td>
<td>0.06</td>
</tr>
<tr>
<td>the</td>
<td>4</td>
<td>0.22</td>
</tr>
<tr>
<td>movie</td>
<td>3</td>
<td>0.17</td>
</tr>
<tr>
<td>actor</td>
<td>2</td>
<td>0.11</td>
</tr>
</tbody>
</table>

**Total**: 18

A new review \( X_{\text{new}} \): **love love the movie**

\[
\log p(X_{\text{new}} \mid \text{positive}) = \sum_{w \in X_{\text{new}}} \log p(w \mid \text{positive}) = -4.96
\]

\[
\log p(X_{\text{new}} \mid \text{negative}) = -8.91
\]
posterior probs for $X_{\text{new}}$

$$p(y \mid x) \propto \arg \max_{y \in Y} p(y) \cdot P(X_{\text{new}} \mid y)$$

$$\log p(\text{positive} \mid X_{\text{new}}) \propto \log P(\text{positive}) + \log p(X_{\text{new}} \mid \text{positive})$$
$$= -0.84 - 4.96 = -5.80$$

$$\log p(\text{negative} \mid X_{\text{new}}) \propto -0.56 - 8.91 = -9.47$$

Naive Bayes predicts a positive label!
what if we see no positive training documents containing the word “awesome”?

\[ p(\text{awesome} \mid \text{positive}) = 0 \]

any review that contains “awesome” will have zero probability for the positive class!
Laplace (add-1) smoothing for Naïve Bayes

\[
\hat{P}(w_i \mid c) = \frac{\text{count}(w_i, c)}{\sum_{w \in V} (\text{count}(w, c))}
\]

\[=
\frac{\text{count}(w_i, c) + 1}{\left( \sum_{w \in V} \text{count}(w, c) \right) + |V|}
\]

what happens if we do add-\(n\) smoothing as \(n\) increases?
exercise!