Text Classification and Naïve Bayes

These slides are based on:

Dan Jurafsky and James H. Martin, Speech and Language Processing (3rd ed. draft)
https://web.stanford.edu/~jurafsky/slp3/
(Chapter 7)

These slides are an edited version of Jurafsky’s slides:
https://web.stanford.edu/~jurafsky/NLPCourseraSlides.html
Examples of text classification problems
Positive or negative movie review?

• unbelievably disappointing
• full of zany characters and richly applied satire, and some great plot twists
• this is the greatest screwball comedy ever filmed
• it was pathetic; the worst part about it was the boxing scenes.
Dear My Friends,

Good day!

Kerric Laboratory Equipment Research & Development Manufacturer Co., Ltd is established since 1999, is the leading manufacturer & supplier of the LABORATORY FURNITURE and relevant accessory for school, college, university and chemical or biology industry.

Our products Range Includes:
Laboratory Bench (All Steel, Steel Wood)
Laboratory Cabinet (All Steel, Aluminum Wood)
Fume Hood (All steel, PP)
Laboratory Rack/Shelf (All Steel, Steel Wood)
What is the topic of this post?

- Agriculture
- Robotics
- Sport
- Religion
- Psychology
- ...

Smithsonian Magazine @SmithsonianMag - Jul 6
The FarmBot Genesis brings precision agriculture to your own backyard via @ModFarm, smithmag.co/INSxK

Cristiano Ronaldo @Cristiano - Jul 8
Time to focus. Counting down the hours with my @TAGHeuer. See you on Sunday on the field. #dontcrackunderpressure
Text Classification

Assigning subject categories:

- Sentiment analysis
- Spam detection
- Language identification
- Authorship identification
- Topic identification
- ...
Text Classification: definition

• **Input:**
  - a document $d$
  - a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$

• **Output:** a predicted class $c \in C$
Classification Methods:
Hand-coded rules and dictionary methods

• Rules based on combinations of words or other features
  o spam: black-list-address OR (“dollars” AND “have been selected”)

• Accuracy can be high
  o If rules carefully refined by expert

• But building and maintaining these rules is expensive
Text Classification: Supervised Machine Learning
Classification Methods: Supervised Machine Learning

• **Input:**
  o a document $d$
  o a fixed set of classes $C = \{c_1, \ldots, c_j\}$
  o A training set of $m$ hand-labeled documents $(d_1, c_1), \ldots, (d_m, c_m)$

• **Output:**
  o a learned classifier $\gamma: d \rightarrow c$
Classification Methods:
Supervised Machine Learning

• Any kind of classifier
  o Naïve Bayes
  o Logistic regression
  o Support-vector machines
  o k-Nearest Neighbors
  o ...

The bag of words (BOW)

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet.

γ( ) = c
I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet.
BOW: using a subset of words

\[ y(\gamma) = c \]

love, satirical, great, fun, whimsical, romantic, laughing, recommend, several, happy, again.
\[ \gamma() = C \]

<table>
<thead>
<tr>
<th>Word</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>great</td>
<td>2</td>
</tr>
<tr>
<td>love</td>
<td>2</td>
</tr>
<tr>
<td>recommend</td>
<td>1</td>
</tr>
<tr>
<td>laugh</td>
<td>1</td>
</tr>
<tr>
<td>happy</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Text Classification: Naïve Bayes
Naïve Bayes Intuition

• Simple (“naïve”) classification method based on Bayes rule
• Relies on very simple representation of document
  o Bag of words or n-grams
Bayes’ Rule applied to documents and classes

For a document $d$ and a class $c$

$$P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)}$$
Naïve Bayes Classifier

\[ c_{MAP} = \arg\max_{c \in C} P(c | d) \]

\[ = \arg\max_{c \in C} \frac{P(d | c)P(c)}{P(d)} \]

Bayes theorem

\[ = \arg\max_{c \in C} P(d | c)P(c) \]

Dropping the denominator

MAP is “maximum a posteriori” = most likely class
Naïve Bayes Classifier

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

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

Features could be:
- words (binary value)
- word counts
- word frequencies (tf)
- tf-idf

Document \( d \) represented as features \( x_1..x_n \)
Naïve Bayes Classifier

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

- \( O(|X|^n \cdot |C|) \) parameters
- How often does this class occur?
- Could only be estimated if a very, very large number of training examples was available.
- We can just count the relative frequencies in a corpus
Multinomial NB: Independence Assumptions

- **Bag of words assumption**: Assume that position of words doesn’t matter (the exchangeability of random variables)

\[ P(x_1, x_2, \ldots, x_n \mid c) = P(x_{\delta(1)}, x_{\delta(2)}, \ldots, x_{\delta(n)} \mid c) \]

- **Conditional Independence**: Assume the feature probabilities \( P(x_i \mid c_j) \) are independent given the class \( c \).

\[ P(x_1, \ldots, x_n \mid c) = P(x_1 \mid c) \cdot P(x_2 \mid c) \cdot P(x_3 \mid c) \cdot \ldots \cdot P(x_n \mid c) \]
Multinomial Naïve Bayes Classifier

Features: counts
Likelihood: \( P(x) = Multinomial(x) \)

\[
c_{NB} = \arg\max_{c \in C} P(c) \prod_{i \in V} P(x_i | c)
\]
Boolean Multinomial Naïve Bayes

Features: binarized counts (0/1 values)
Likelihood: \( P(x) = \text{Multinomial}(x) \)

\[
c_{NB} = \arg\max_{c \in C} P(c) \prod_{i \in V} P(x_i | c)
\]

- Boolean (aka binarized) multinomial NB good for polarity prediction
- Different from Bernoulli Naïve Bayes classifier
Bernoulli Naïve Bayes Classifier

Features: binarized counts (0/1 values)
Likelihood: \( P(x) = \text{MultivariateBernoulli}(x) \)

\[
c_{NB} = \underset{c \in C}{\text{argmax}} \ P(c) \prod_{i \in V} \left( P(x_i | c) \right)^{x_i} (1 - P(x_i | c))^{1-x_i}
\]
Learning parameters of Naïve Bayes
Learning parameters of Multinomial NB

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

\[
\hat{P}(c_j) = \frac{\text{doccount}(C = c_j)}{N_{doc}}
\]

\[
\hat{P}(w_i | c_j) = \frac{\text{count}(w_i, c_j)}{\sum_{w \in V} \text{count}(w, c_j)}
\]
Parameter estimation

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

fraction of times word \( w_i \) appears among all words in documents of topic \( c_j \)

- Create mega-document for topic \( j \) by concatenating all docs in this topic
  - Use the frequency of \( w \) in mega-document
Problem with Maximum Likelihood

- What if we have seen no training documents with the word fantastic and classified in the topic positive (thumbs-up)?

\[ \hat{P}(\text{fantastic} \mid \text{positive}) = \frac{\text{count}(\text{fantastic}, \text{positive})}{\sum_{w \in V} \text{count}(w, \text{positive})} = 0 \]

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

\[ c_{MAP} = \arg\max_c \hat{P}(c) \prod_i \hat{P}(x_i \mid c) \]
Laplace (add-1) smoothing for Naïve Bayes

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

Additive smoothing:

\[
\hat{P}(w_i \mid c) = \frac{\text{count}(w_i, c) + \alpha}{\left( \sum_{w \in V} \text{count}(w, c) \right) + \alpha |V|}
\]
Multinomial Naïve Bayes: Learning

• From training corpus, extract *Vocabulary*

• Calculate $P(c_j)$ terms
  - For each $c_j$ in $C$ do
    - $docs_j \leftarrow$ all docs with class $= c_j$
    - $P(c_j) \leftarrow \frac{|docs_j|}{|\text{total # documents}|}$

• Calculate $P(w_k \mid c_j)$ terms
  - $Text_j \leftarrow$ single doc containing all $docs_j$
  - For each word $w_k$ in *Vocabulary*
    - $n_k \leftarrow$ # of occurrences of $w_k$ in $Text_j$
    - $P(w_k \mid c_j) \leftarrow \frac{n_k + \alpha}{n + \alpha |\text{Vocabulary}|}$
Summary: Naive Bayes surprisingly good

- Very Fast, low storage requirements
- Robust to Irrelevant Features
  Irrelevant Features cancel each other without affecting results
- Very good in domains with many equally important features
  Decision Trees suffer from *fragmentation* in such cases – especially if little data
- Optimal if the independence assumptions hold: If assumed independence is correct, then it is the Bayes Optimal Classifier for problem
- A good dependable baseline for text classification
  - But we will see other classifiers that give better accuracy
Naïve Bayes in Spam Filtering

- **SpamAssassin Features:**
  - Mentions Generic Viagra
  - Online Pharmacy
  - Mentions millions of (dollar) ((dollar) NN,NNN,NNN.NN)
  - Phrase: impress ... girl
  - From: starts with many numbers
  - Subject is all capitals
  - HTML has a low ratio of text to image area
  - One hundred percent guaranteed
  - Claims you can be removed from the list
  - 'Prestigious Non-Accredited Universities'
  - [http://spamassassin.apache.org/tests_3_3_x.html](http://spamassassin.apache.org/tests_3_3_x.html)
Evaluation metrics: precision, recall, accuracy,
The 2-by-2 contingency table

<table>
<thead>
<tr>
<th></th>
<th>Actual class:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>positive</td>
</tr>
<tr>
<td>predicted class:</td>
<td></td>
</tr>
<tr>
<td>positive</td>
<td>tp</td>
</tr>
<tr>
<td>negative</td>
<td>fn</td>
</tr>
</tbody>
</table>

true negative
**Precision and recall**

- **Precision**: % of predicted positive items that are correctly classified
- **Recall**: % of actual positive items that are correctly classified
- **Accuracy**: % of all items that are correctly classified

<table>
<thead>
<tr>
<th>Predicted class:</th>
<th>positive</th>
<th>negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>positive</td>
<td>tp</td>
<td>fp</td>
</tr>
<tr>
<td>negative</td>
<td>fn</td>
<td>tn</td>
</tr>
</tbody>
</table>

**Formulas**:

- \[ P = \frac{tp}{tp + fp} \]
- \[ R = \frac{tp}{tp + fn} \]
- \[ A = \frac{tp + tn}{n + p} \]
Confusion matrix $c_{ij}$

<table>
<thead>
<tr>
<th>Actual class</th>
<th>Predicted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cat</td>
</tr>
<tr>
<td>Cat</td>
<td>5</td>
</tr>
<tr>
<td>Dog</td>
<td>2</td>
</tr>
<tr>
<td>Rabbit</td>
<td>0</td>
</tr>
</tbody>
</table>

$c_{33} = 11$

i.e., 11 rabbits correctly classified as rabbits

$c_{32} = 2$

i.e., 2 rabbits incorrectly classified as dogs
Per class evaluation measures

Recall:
Fraction of docs in class $i$ classified correctly:

\[
R = \frac{C_{ii}}{\sum_j C_{ij}}
\]

Precision:
Fraction of docs assigned class $i$ that are actually about class $i$:

\[
P = \frac{C_{ii}}{\sum_j C_{ji}}
\]

Accuracy: (1 - error rate)
Fraction of docs classified correctly:

\[
Accuracy = \frac{\sum_i C_{ii}}{\sum_j \sum_i C_{ij}}
\]
A combined measure: $F$

- A combined measure that assesses the P/R tradeoff is $F$ measure (weighted harmonic mean):

\[
F = \frac{1}{\alpha \frac{1}{P} + (1 - \alpha) \frac{1}{R}}
\]

- The harmonic mean is a very conservative average
- We typically use balanced $F$ measure with $\alpha = 0.5$
  - Namely, $F_1 = \frac{2PR}{P+R}$
Cross-validation

Data:
- Training set
- Development Test (Validation) Set
- Test Set

- **Score metric:** P/R/F1 or Accuracy
- **Unseen test set**
  - avoid overfitting (‘tuning to the test set’)
  - more conservative estimate of performance
- **Cross-validation over multiple splits**
  - compute score metric for each split
  - average score over all splits
- **Grid search over hyperparameters**
  - repeat previous step for various values of hyperparameters to find the best ones
  - choose hyperparameters that maximize score

CV, 3 splits (folds):
- Training Set
- Dev Test
- Training Set
- Dev Test
- Dev Test
- Training Set
- Test Set
Text Classification and Naïve Bayes

These slides are based on:

Dan Jurafsky and James H. Martin, Speech and Language Processing (3rd ed. draft)
https://web.stanford.edu/~jurafsky/slp3/
(Chapter 7)

These slides are an edited version of Jurafsky’s slides:
https://web.stanford.edu/~jurafsky/NLPCourseraSlides.html