Text classification: The Naive Bayes Classifier

Chapter 4 in Martin/Jurafsky

Is this spam?

Investing in Films Offers Potential of high Returns | Starting GBP 10,000

Blockbuster Films <no-reply@luxuryapartments.com>

Why is this message in spam? It is similar to messages that were identified as spam in the past.

Report not spam
Who wrote which Federalist papers?

- 1787-8: anonymous essays try to convince New York to ratify U.S Constitution: Jay, Madison, Hamilton.
- Authorship of 12 of the letters in dispute
- 1963: solved by Mosteller and Wallace using Bayesian methods

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.
What is the subject of this article?

MEDLINE article

MeSH Subject Category Hierarchy
- Antagonists and Inhibitors
- Blood Supply
- Chemistry
- Drug Therapy
- Embryology
- Epidemiology

Text classification
- Assigning subject categories, topics, or genres
- Spam detection
- Authorship identification
- Age/gender identification
- Language identification
- Sentiment analysis
- ...
Text classification: problem definition

• Input:
  – a document \( d \)
  – a fixed set of classes \( C = \{c_1, c_2, \ldots, c_j\} \)

• Output: a predicted class \( c \in C \) (and a level of confidence in the prediction)

Classification methods: Hand-coded rules

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

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

• Difficulty?
Classification methods: Hand-coded rules

• Rules based on combinations of words or other features
  – spam: black-list-address OR (“dollars” AND “have been selected”)
• Accuracy can be high
  – If rules carefully refined by expert
• Building and maintaining these rules is time-consuming

Classification using supervised machine learning

• Input:
  – a document d
  – a fixed set of classes \( C = \{c_1, c_2, \ldots, c_J\} \)
  – A training set of N hand-labeled documents \( (d_1, c_1), \ldots, (d_N, c_N) \)
• Output:
  – a learned classifier which is a mapping from the set of documents to the set of labels
Classification methods: Supervised machine learning

• Any kind of classifier can be used for this task:
  – Naïve Bayes
  – Logistic regression
  – Support-vector machines
  – Neural networks
  – …

Naïve Bayes Intuition

• Simple (“naïve”) classification method based on Bayes rule
• Relies on very simple representation of document
  – Bag of words
The "bag of words" representation

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!

Classification with probabilistic models

- To classify a document \( d \) choose the class that has the highest probability:

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

From the product rule:
\[ P(x, y) = P(y|x) P(x) \]
and:
\[ P(x, y) = P(x|y) P(y) \]

Therefore:
\[ P(y|x) = \frac{P(x|y)P(y)}{P(x)} \]

This is known as Bayes’ rule

Bayes’ rule for classification of documents

• For a document \( d \) and a class \( c \)

\[ P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)} \]
MAP classification

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

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

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

MAP is “maximum a posteriori” = most likely class

Bayes Rule

Dropping the denominator

MAP classification

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

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

Document d represented as features \( x_1, \ldots, x_n \)
MAP classification

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

- 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.

Naïve Bayes independence assumption

$$P(x_1, x_2, \ldots, x_n \mid c)$$

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

$$P(x_1, \ldots, x_n \mid c) = P(x_1 \mid c)P(x_2 \mid c)P(x_3 \mid c), \ldots, P(x_n \mid c)$$
Multinomial Naïve Bayes

- **Conditional Independence**: Assume the feature probabilities $P(x_i | c_j)$ are independent given the class $c$.
  
  $$P(x_1, \ldots, x_n | c) = P(x_1 | c) P(x_2 | c) P(x_3 | c), \ldots, P(x_n | c)$$

- **Bag of Words assumption**: position doesn’t matter; the variables represent counts/presence absence of a word in a document

Multinomial Naïve Bayes Classifier

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

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

How many parameters in a model with a vocabulary $V$?
Learning the Multinomial Naïve Bayes Model

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

\[
\hat{P}(c) = \frac{N_c}{N_{doc}} \quad \text{fraction of documents in class c}
\]

\[
\hat{P}(w_i | c) = \frac{\text{count}(w_i, c)}{\sum_{w \in V} \text{count}(w, c)} \quad \text{fraction of times word } w \text{ appears among all words in documents of topic c}
\]

V – the vocabulary

Maximum likelihood

• Fit a probabilistic model \( P(x | \theta) \) to data
  – Estimate \( \theta \)

• Given independent identically distributed (i.i.d.) data \( X = (x_1, x_2, \ldots, x_n) \)
  – Likelihood

\[
P(X | \theta) = P(x_1 | \theta)P(x_2 | \theta), \ldots, P(x_n | \theta)
\]

  – Log likelihood

\[
\ln P(X | \theta) = \sum_{i=1}^{n} \ln P(x_i | \theta)
\]

• Maximum likelihood solution: parameters \( \theta \) that maximize \( \ln P(X | \theta) \)
Example

- Example: coin toss
  - Estimate the probability $p$ that a coin lands “Heads” using the result of $n$ coin tosses, $h$ of which resulted in heads.
  - The likelihood of the data: $P(X|\theta) = p^h (1-p)^{n-h}$
  - Log likelihood: $\ln P(X|\theta) = h \ln p + (n-h) \ln(1-p)$
  - Taking a derivative and setting to 0: $\frac{\partial \ln P(X|\theta)}{\partial p} = \frac{h}{p} - \frac{(n-h)}{(1-p)} = 0$
    $$\Rightarrow p = \frac{h}{n}$$

Parameter estimation

$$\hat{P}(w_i | c) = \frac{\text{count}(w_i, c)}{\sum_{w \in V} \text{count}(w, c)}$$

- Create mega-document for topic $c$ by concatenating all docs in this topic
  - Use frequency of $w$ in mega-document
Problem with zeros

- What if we have seen no training documents with the word \textit{fantastic} and classified in the topic \textbf{positive} (\textit{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!

\[ c_{\text{MAP}} = \arg\max_c \hat{P}(c) \prod_i \hat{P}(x_i \mid c) \]

Laplace (add-1) smoothing for Naïve Bayes

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

- From training corpus, extract vocabulary \( V \)

- Calculate \( P(c) \) terms

\[
P(c) \leftarrow \frac{N_c}{N_{doc}}
\]

- Calculate \( P(w_k \mid c) \) terms
  - For each word \( w_k \) in the vocabulary \( V \):
    \[
    n_k \leftarrow \# \text{ of occurrences of } w_k \text{ in documents in class } c
    \]
    \[
    n \leftarrow \text{total number of words in documents in class } c
    \]
    \[
P(w_k \mid c) \leftarrow \frac{n_k + \alpha}{n + \alpha |V|}
    \]

<table>
<thead>
<tr>
<th>Doc</th>
<th>Words</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>1 Chinese Beijing Chinese</td>
<td>c</td>
</tr>
<tr>
<td></td>
<td>2 Chinese Chinese Shanghai</td>
<td>c</td>
</tr>
<tr>
<td></td>
<td>3 Chinese Macao</td>
<td>c</td>
</tr>
<tr>
<td></td>
<td>4 Tokyo Japan Chinese</td>
<td>j</td>
</tr>
<tr>
<td>Test</td>
<td>5 Chinese Chinese Chinese Tokyo Japan</td>
<td>?</td>
</tr>
</tbody>
</table>

Priors:
- \( P(c) = \frac{3}{4} \)
- \( P(j) = \frac{1}{4} \)

Conditional Probabilities:
- \( P(\text{Chinese} \mid c) = \frac{5+1}{8+6} = \frac{6}{14} = \frac{3}{7} \)
- \( P(\text{Tokyo} \mid c) = \frac{0+1}{8+6} = \frac{1}{14} \)
- \( P(\text{Japan} \mid c) = \frac{0+1}{8+6} = \frac{1}{14} \)
- \( P(\text{Chinese} \mid j) = \frac{1+1}{3+6} = \frac{2}{9} \)
- \( P(\text{Tokyo} \mid j) = \frac{1+1}{3+6} = \frac{2}{9} \)
- \( P(\text{Japan} \mid j) = \frac{1+1}{3+6} = \frac{2}{9} \)

Making a prediction:
- \( P(c \mid d5) \propto \frac{3/4 \ast (3/7)^3 \ast 1/14 \ast 1/14}{1/4} \approx 0.0003 \)
- \( P(j \mid d5) \propto \frac{1/4 \ast (2/9)^3 \ast 2/9 \ast 2/9}{2/9} \approx 0.0001 \)
Multinomial Naïve Bayes as a generative model

\[ e = \text{China} \]

\[ X_1 = \text{Shanghai} \]
\[ X_2 = \text{and} \]
\[ X_3 = \text{Shenzhen} \]
\[ X_4 = \text{issue} \]
\[ X_5 = \text{bonds} \]

Naïve Bayes and Language Modeling

- Naïve Bayes classifiers can use any sort of feature
  - URL, email address, network features etc.
- But if, as in the previous slides
  - We use only word features
- Then
  - Naïve Bayes is related to language models.
Each class is a unigram language model

- Assigning each word: \( P(\text{word} \mid c) \)
- Assigning each sentence: \( P(s \mid c) = \prod P(\text{word} \mid c) \)

Class \( \text{pos} \)

<table>
<thead>
<tr>
<th>0.1</th>
<th>I</th>
<th></th>
<th>love</th>
<th>this</th>
<th>fun</th>
<th>film</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>love</td>
<td></td>
<td>0.1</td>
<td>0.1</td>
<td>0.05</td>
<td>0.01</td>
</tr>
<tr>
<td>0.01</td>
<td>this</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1</td>
</tr>
<tr>
<td>0.05</td>
<td>fun</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>film</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\( P(s \mid \text{pos}) = 0.0000005 \)

Naïve Bayes as a Language Model

- Which class assigns the higher probability to \( s \)?

Model \( \text{pos} \)

<table>
<thead>
<tr>
<th>0.1</th>
<th>I</th>
<th></th>
<th>love</th>
<th>this</th>
<th>fun</th>
<th>film</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>love</td>
<td></td>
<td>0.001</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td>this</td>
<td></td>
<td>0.01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.05</td>
<td>fun</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>film</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Model \( \text{neg} \)

<table>
<thead>
<tr>
<th>0.2</th>
<th>I</th>
<th></th>
<th>love</th>
<th>this</th>
<th>fun</th>
<th>film</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>love</td>
<td></td>
<td>0.001</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td>this</td>
<td></td>
<td>0.01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.005</td>
<td>fun</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>film</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\( P(s \mid \text{pos}) > P(s \mid \text{neg}) \)
Naïve Bayes in spam filtering

- SpamAssassin has a version that uses Naive Bayes and uses a variety of features of email messages:
  - Mentions Viagra (or other drugs)
  - Online pharmacy
  - Mentions millions of (dollar) ((dollar) NN,NNN,NNN,NN)
  - From: starts with many numbers
  - Subject is all capitals
  - Claims you can be removed from the list
  - https://spamassassin.apache.org/old/tests_3_3_x.htm

Naïve Bayes Summary

- Very fast; low storage requirements
- Robust to irrelevant features
  - Irrelevant features cancel each other without affecting results
- Optimal if the independence assumptions hold
- A good dependable baseline for text classification
  - But we will see other classifiers that give better accuracy