Is this spam?

---------- Forwarded message ----------
From: E-Lotto <infoalert@wp.pl>
Date: Fri, Sep 16, 2016 at 3:24 PM
Subject: Ref#.EL16BDXAUL16-03
To:

Hi,

E-Lotto congratulates you as the winner of $2,500,000.00. Email Rep (Aaron Martins) at aaronmarts@excite.com with Ref#.EL16BDXAUL20

- Text Classification
  - Naive Bayes Model of Documents
    \[ \Rightarrow \]
    Naive Bayes Classifier
    \[ \Leftarrow \]
    Class-Conditional LM
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

James Madison  
Alexander Hamilton
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
- ...

Terms of Serviceinterp.
- Have arb. cur?
- False News vs Real News

Language ID
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

- 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
- But building and maintaining these rules is expensive

→ High coverage is hard

- Human bias
- Alternate spellings
- Problem drift... data distribution changes
Classification Methods: Supervised Machine Learning

**Input:**
- a document \( d \)
- a fixed set of classes \( C = \{c_1, c_2, \ldots, c_i\} \)
- A training set of \( m \) hand-labeled documents \( (d_1, c_1), \ldots, (d_m, c_m) \)

**Output:**
- a learned classifier \( f: d \rightarrow c \)

![Diagram](Diagram.png)
Sup. Learning Algos?

- Decision Trees
- K-NN
- Linear Regression
- Logistic Regression
- SVMs
- Neural Networks
- Naïve Bayes
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!

The Bag of Words Representation

<table>
<thead>
<tr>
<th>Word</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>it</td>
<td>6</td>
</tr>
<tr>
<td>I</td>
<td>5</td>
</tr>
<tr>
<td>the</td>
<td>4</td>
</tr>
<tr>
<td>to</td>
<td>3</td>
</tr>
<tr>
<td>and</td>
<td>3</td>
</tr>
<tr>
<td>seen</td>
<td>2</td>
</tr>
<tr>
<td>yet</td>
<td>1</td>
</tr>
<tr>
<td>would</td>
<td>1</td>
</tr>
<tr>
<td>whimsical</td>
<td>1</td>
</tr>
<tr>
<td>times</td>
<td>1</td>
</tr>
<tr>
<td>sweet</td>
<td>1</td>
</tr>
<tr>
<td>satirical</td>
<td>1</td>
</tr>
<tr>
<td>adventure</td>
<td>1</td>
</tr>
<tr>
<td>genre</td>
<td>1</td>
</tr>
<tr>
<td>fairy</td>
<td>1</td>
</tr>
<tr>
<td>humor</td>
<td>1</td>
</tr>
<tr>
<td>have</td>
<td>1</td>
</tr>
<tr>
<td>great</td>
<td>1</td>
</tr>
</tbody>
</table>

Problems

- Narrative ordering
- Avg. ordering

- Negations
- Compositional meaning
Generative Model

doc categ

Assume process: \( \mathbb{LM} \) \( P(d|lc) \)

\[ P(c|d) = \frac{P(d|lc) P(c)}{P(d)} \]

Normalizer

\[ = \sum_{c \in C} P(d|lc) P(c) \]
Naïve Bayes Classifier (I)

$$c_{MAP} = \underset{c \in C}{\text{argmax}} \ P(c \mid d)$$

$$= \underset{c \in C}{\text{argmax}} \ \frac{P(d \mid c)P(c)}{P(d)}$$

$$= \underset{c \in C}{\text{argmax}} \ P(d \mid c)P(c)$$

MAP is "maximum a posteriori" = most likely class

Bayes Rule

Dropping the denominator

$$\max_{x \in \{1, 2, 3\}} x^2 = 9$$

$$\min_{x \in \{1, 2, 3\}} x^2 = 1$$
Multinomial Naïve Bayes Classifier

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

\[ c_{\text{NB}} = \arg\max_{c \in C} P(c_j) \prod_{x \in X} P(x \mid c) \]
Multinomial Naïve Bayes Independence Assumptions

\[ P(d \mid c) = P(x_1, x_2, \ldots, x_n \mid c) \]

- **Bag of Words assumption**: Assume position doesn’t matter
- **Conditional Independence**: Assume the feature probabilities \( P(x_i \mid c) \) are independent given the class \( c \).

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

\[ \text{Can estimate!} \]
Applying Multinomial Naive Bayes Classifiers to Text Classification

positions ← all word positions in test document

\[
P(c_3) \quad P(d \mid c_3)
\]

\[
c_{NB} = \text{argmax} \quad P(c_j) \prod_{c_j \in C} P(x_i \mid c_j)
\]

\[
\prod_{i \in \text{positions}}
\]
Naïve Bayes as a Language Model

- Which class assigns the higher probability to s?

\[
P(\text{pos} | d) = \frac{P(d | \text{pos}) \cdot P(\text{pos})}{P(d)}\]

\[
P(\text{neg} | d) = \frac{P(d | \text{neg}) \cdot P(\text{neg})}{P(d)}\]

\[
P(d | \text{pos}) = P(\text{I} | \text{pos}) \cdot P(\text{love} | \text{pos}) \cdot P(\text{this} | \text{pos}) \cdot P(\text{fun} | \text{pos}) \cdot P(\text{film} | \text{pos})
\]

\[
P(d | \text{neg}) = P(\text{I} | \text{neg}) \cdot P(\text{love} | \text{neg}) \cdot P(\text{this} | \text{neg}) \cdot P(\text{fun} | \text{neg}) \cdot P(\text{film} | \text{neg})
\]

\[
P(d | \text{pos}) \geq P(d | \text{neg})
\]

"Likelihood Ratio"
Learning the Multinomial Naïve Bayes Model

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

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

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

*Num tokens of all docs in $c_j$*
Problem with Maximum Likelihood

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

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

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

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

0 !!! BAD
Laplace (add-1) smoothing for Naïve Bayes

\[
\hat{P}(w_i | c) = \frac{\text{count}(w_i, c) + 1}{\sum_{w \in V} (\text{count}(w, c) + 1)} \quad -\alpha \\
\quad + \frac{\alpha}{|V|}
\]

"Pseudo count smoothing"

"Add-\alpha smoothing"
Multinomial Naïve Bayes: Learning

- From training corpus, extract Vocabulary

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

- Calculate $P(w_k | c_j)$ 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 $Text_j$
    - $P(w_k | c_j) \leftarrow \frac{n_k + \alpha}{n + \alpha |\text{Vocabulary}|}$
Generative and Discriminative Text Classification with Recurrent Neural Networks

Dani Yogatama, Chris Dyer, Wang Ling, and Phil Blunsom
DeepMind
{dyogatama, cdyer, lingwang, pblunsom}@google.com

Abstract

We empirically characterize the performance of discriminative and generative LSTM models for text classification. We find that although RNN-based generative models are more powerful than their bag-of-words ancestors (e.g., they account for conditional dependencies across words in a document), they have higher

Table 2: Summary of results on the full datasets.

<table>
<thead>
<tr>
<th>Models</th>
<th>AGNews</th>
<th>Sogou</th>
<th>Yelp Bin</th>
<th>Yelp Full</th>
<th>DBPedia</th>
<th>Yahoo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive Bayes</td>
<td>90.0</td>
<td>86.3</td>
<td>86.0</td>
<td>51.4</td>
<td>96.0</td>
<td>68.7</td>
</tr>
<tr>
<td>Kneser-Ney Bayes</td>
<td>89.3</td>
<td>94.6</td>
<td>81.8</td>
<td>41.7</td>
<td>95.4</td>
<td>69.3</td>
</tr>
<tr>
<td>MLP Naive Bayes</td>
<td>89.9</td>
<td>76.1</td>
<td>73.6</td>
<td>40.4</td>
<td>87.2</td>
<td>60.6</td>
</tr>
<tr>
<td>Discriminative LSTM</td>
<td>92.1</td>
<td>94.9</td>
<td>92.6</td>
<td>59.6</td>
<td>98.7</td>
<td>73.7</td>
</tr>
<tr>
<td>Generative LSTM-independent comp.</td>
<td>90.7</td>
<td>93.5</td>
<td>90.0</td>
<td>51.9</td>
<td>94.8</td>
<td>70.5</td>
</tr>
<tr>
<td>Generative LSTM-shared comp.</td>
<td>90.6</td>
<td>90.3</td>
<td>88.2</td>
<td>52.7</td>
<td>95.4</td>
<td>69.3</td>
</tr>
<tr>
<td>bag of words (Zhang et al., 2015)</td>
<td>88.8</td>
<td>92.9</td>
<td>92.2</td>
<td>58.0</td>
<td>96.6</td>
<td>68.9</td>
</tr>
<tr>
<td>fastText (Joulin et al., 2016)</td>
<td>92.5</td>
<td>96.8</td>
<td>95.7</td>
<td>63.9</td>
<td>98.6</td>
<td>72.3</td>
</tr>
<tr>
<td>char-CNN (Zhang et al., 2015)</td>
<td>87.2</td>
<td>95.1</td>
<td>94.7</td>
<td>62.0</td>
<td>98.3</td>
<td>71.2</td>
</tr>
<tr>
<td>char-CRN (Xiao &amp; Cho, 2016)</td>
<td>91.4</td>
<td>95.2</td>
<td>94.5</td>
<td>61.8</td>
<td>98.6</td>
<td>71.7</td>
</tr>
<tr>
<td>very deep CNN (Conneau et al., 2016)</td>
<td>91.3</td>
<td>96.8</td>
<td>95.7</td>
<td>64.7</td>
<td>98.7</td>
<td>73.4</td>
</tr>
</tbody>
</table>
Summary: Naive Bayes is Not So Naive

- 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