

Naive Bayes Classification (and Sentiment)

Chapter 4

Text Classification

Is this spam?

RE: INVESTMENT/BUSINESS PARTNERSHIP.

DOES YOUR BUSINESS/PROJECT STILL NEED FUNDING?

DO YOU HAVE PROJECT/INVESTMENT CAPABLE OF GENERATING 15% AROI?

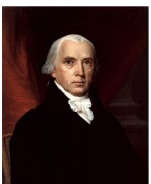
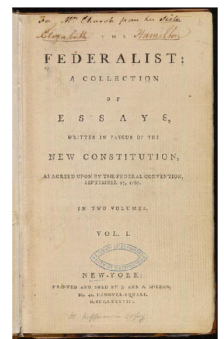
IF THE ANSWER IS YES, I represent a group of company based in the Gulf Region. We are seeking means of expanding and relocating our business interest abroad in the following sector, Oil & Gas, Energy, Mining, Construction, Real Estate, Communication, Agriculture, Health Sector or any other VIABLE business/project capable of generating 15% AROI.

If you have a solid background and idea of making good profit in any of the following SECTORS, please write me for possible business co-operation. More so, we are ready to facilitate and fund any business that is capable of generating 15% Annual Return on Investment (AROI) Joint Venture partnership and Hard loan funding can also be considered.

I look forward to discussing this opportunity further with you.

Who wrote which Federalist papers?

- 1787-8: anonymous essays try to convince New York to ratify U.S Constitution.
- 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.

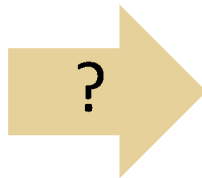
What is the subject of this article?

MEDLINE Article



MeSH Subject Category Hierarchy

- Antagonists and Inhibitors
- Blood Supply
- Chemistry
- Drug Therapy
- Embryology
- Epidemiology
- ...



Text Classification

- Assigning subject categories, topics, or genres
- Spam detection
- Authorship identification
- Age/gender identification
- Language Identification
- Sentiment analysis
- ...

Text Classification: definition

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

Classification Methods: Supervised Machine Learning

- *Input:*
 - a document d
 - a fixed set of classes $C = \{c_1, c_2, \dots, c_j\}$
 - A training set of m hand-labeled documents $(d_1, c_1), \dots, (d_m, c_m)$
- *Output:*
 - a learned classifier $\gamma: d \mapsto c$

Classification Methods: Supervised Machine Learning

- Any kind of classifier
 - Naive Bayes
 - Logistic regression
 - Support-vector machines
 - k-Nearest Neighbors
 - Neural Nets
- ...

Naive Bayes

Intuition and
Formalization

Naive Bayes Intuition

- Simple (“naive”) 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!





| | |
|-----------|-----|
| it | 6 |
| I | 5 |
| the | 4 |
| to | 3 |
| and | 3 |
| seen | 2 |
| yet | 1 |
| would | 1 |
| whimsical | 1 |
| times | 1 |
| sweet | 1 |
| satirical | 1 |
| adventure | 1 |
| genre | 1 |
| fairy | 1 |
| humor | 1 |
| have | 1 |
| great | 1 |
| ... | ... |

The bag of words representation

Y (

| | |
|-----------|-----|
| seen | 2 |
| sweet | 1 |
| whimsical | 1 |
| recommend | 1 |
| happy | 1 |
| ... | ... |

) = C

Bayes' Rule Applied to Documents and Classes

- For a document d and a class c

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)}$$

Naive Bayes Classifier

$$C_{MAP} = \operatorname{argmax}_{c \in C} P(c | d)$$

MAP is "maximum a posteriori" = most likely class

$$= \operatorname{argmax}_{c \in C} \frac{P(d | c)P(c)}{P(d)}$$

Bayes Rule

$$= \operatorname{argmax}_{c \in C} P(d | c)P(c)$$

Dropping the denominator

Naive Bayes Classifier

$$C_{MAP} = \operatorname{argmax}_{c \in C} P(d | c)P(c)$$

$$= \operatorname{argmax}_{c \in C} P(x_1, x_2, \dots, x_n | c)P(c)$$

Document d represented as features $x_1 \dots x_n$

Naive Bayes Classifier

$$C_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(x_1, x_2, \dots, x_n | c) P(c)$$

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

Naive Bayes Independence Assumptions

$$P(x_1, x_2, \dots, x_n | c)$$

- **Bag of Words assumption:** Assume position doesn't matter
- **Conditional Independence:** Assume the feature probabilities $P(x_i | c_j)$ are independent given the class c .

$$P(x_1, \dots, x_n | c) = P(x_1 | c) \cdot P(x_2 | c) \cdot P(x_3 | c) \cdot \dots \cdot P(x_n | c)$$

Naive Bayes Classifier

$$C_{MAP} = \operatorname{argmax}_{c \in C} P(x_1, x_2, \dots, x_n | c) P(c)$$

$$C_{NB} = \operatorname{argmax}_{c \in C} P(c_j) \prod_{x \in X} P(x | c)$$

Applying Naive Bayes to Text Classification

positions \leftarrow all word positions in test document

$$C_{NB} = \operatorname{argmax}_{c_j \in C} P(c_j) \prod_{i \in \text{positions}} P(x_i | c_j)$$

Naive Bayes

Learning

Sec. 13.3

Learning the Naive Bayes Model

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

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

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

Parameter estimation

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

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 frequency of w in mega-document

lec 13.3

Problem with Maximum Likelihood

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

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

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

$$c_{MAP} = \operatorname{argmax}_c \hat{P}(c) \prod_i \hat{P}(x_i | c)$$

Laplace (add-1) smoothing for Naive Bayes

$$\hat{P}(w_i | c) = \frac{\text{count}(w_i, c) + 1}{\sum_{w \in V} (\text{count}(w, c) + 1)}$$

$$= \frac{\text{count}(w_i, c) + 1}{\sum_{w \in V} \text{count}(w, c) + |V|}$$

Naive Bayes: Learning

- From training corpus, extract *Vocabulary*
- Calculate $P(c_j)$ terms
 - For each c_j in C do
 - $\text{docs}_j \leftarrow$ all docs with class = c_j
 - $P(c_j) \leftarrow \frac{|\text{docs}_j|}{|\text{total \# documents}|}$
- Calculate $P(w_k | c_j)$ terms
 - $\text{Text}_j \leftarrow$ single doc containing all docs_j
 - For each word w_k in *Vocabulary*
 - $n_k \leftarrow$ # of occurrences of w_k in Text_j
 - $P(w_k | c_j) \leftarrow \frac{n_k + \alpha}{n + \alpha | \text{Vocabulary} |}$

Naive Bayes

Relationship to Language Modeling

Generative vs Discriminative Classifiers

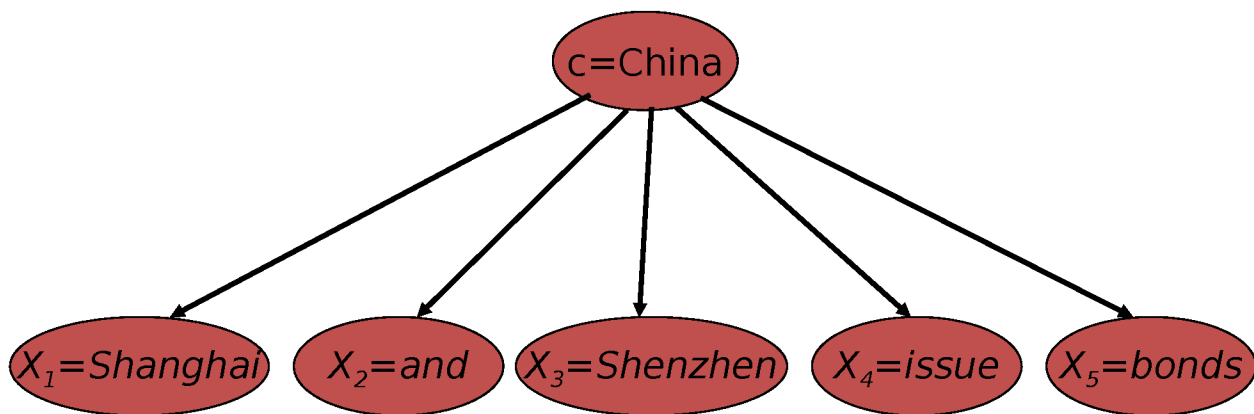
Naive Bayes is the prototypical generative classifier.

- It describes a probabilistic process – “generative story” for a text input X
- But why model X ? It's always observed.

Discriminative models instead:

- seek to optimize a performance measure, like accuracy
- do not worry about $p(X)$

Generative Model for Naive Bayes



31

Naive Bayes and Language Modeling

- Naive bayes classifiers can use any sort of feature
 - URL, email address, dictionaries, network features
- But if, as in the previous slides
 - We use **only** word features
 - we use **all** of the words in the text (not a subset)
- Then
 - Naive bayes has an important similarity to language modeling.

32

Each class = a unigram language model

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

Class *pos*

| | | | | | | | |
|-----------|--|----------|-------------|-------------|------------|-------------|---------------------------------|
| 0.1 I | | <u>I</u> | <u>love</u> | <u>this</u> | <u>fun</u> | <u>film</u> | |
| 0.1 love | | | | | | | |
| 0.01 this | | 0.1 | 0.1 | .01 | .05 | 0.1 | |
| 0.05 fun | | | | | | | |
| 0.1 film | | | | | | | |
| ... | | | | | | | $P(s \text{pos}) = 0.0000005$ |

Naïve Bayes as a Language Model

- Which class assigns the higher probability to *s*?

| | | | | | | | | | | | | | | | | | | | | |
|---|--|---|-------------|------------|-------------|-------------|------------|-------------|--|-----|-----|------|------|-----|--|-----|-------|------|-------|-----|
| <p style="text-align: center; color: green;">Model <i>pos</i></p> <p>0.1 I</p> <p>0.1 love</p> <p>0.01 this</p> <p>0.05 fun</p> <p>0.1 film</p> | <p style="text-align: center; color: red;">Model <i>neg</i></p> <p>0.2 I</p> <p>0.001 love</p> <p>0.01 this</p> <p>0.005 fun</p> <p>0.1 film</p> | <table> <tr> <td></td> <td><u>I</u></td> <td><u>love</u></td> <td><u>this</u></td> <td><u>fun</u></td> <td><u>film</u></td> </tr> <tr> <td></td> <td>0.1</td> <td>0.1</td> <td>0.01</td> <td>0.05</td> <td>0.1</td> </tr> <tr> <td></td> <td>0.2</td> <td>0.001</td> <td>0.01</td> <td>0.005</td> <td>0.1</td> </tr> </table> <p style="text-align: right;">$P(s \text{pos}) > P(s \text{neg})$</p> | | <u>I</u> | <u>love</u> | <u>this</u> | <u>fun</u> | <u>film</u> | | 0.1 | 0.1 | 0.01 | 0.05 | 0.1 | | 0.2 | 0.001 | 0.01 | 0.005 | 0.1 |
| | <u>I</u> | <u>love</u> | <u>this</u> | <u>fun</u> | <u>film</u> | | | | | | | | | | | | | | | |
| | 0.1 | 0.1 | 0.01 | 0.05 | 0.1 | | | | | | | | | | | | | | | |
| | 0.2 | 0.001 | 0.01 | 0.005 | 0.1 | | | | | | | | | | | | | | | |

Generative vs Discriminative Classifiers

- Generative
 - Probabilistic
 - Specify a joint probability distribution over observations and targets: $P(c,d)$
 - Bayes rule enables a conditional distribution
- Discriminative
 - Provide a model for the target variable
 - Use analysis of observed variables
 - Learn boundaries between classes
 - Infer outputs based on inputs: $P(c|d)$

35

Naive Bayes

A Worked Example

$$\hat{P}(c) = \frac{N_c}{N}$$

$$\hat{P}(w|c) = \frac{\text{count}(w,c) + 1}{\text{count}(c) + |V|}$$

| | Doc | Words | Class |
|----------|-----|-------------------------------------|-------|
| Training | 1 | Chinese Beijing Chinese | c |
| | 2 | Chinese Chinese Shanghai | c |
| | 3 | Chinese Macao | c |
| | 4 | Tokyo Japan Chinese | j |
| Test | 5 | Chinese Chinese Chinese Tokyo Japan | ? |

Priors:

$$P(c) = \frac{3}{4} \quad \frac{1}{4}$$

$$P(j) =$$

Choosing a class:

$$P(c|d5) \propto \frac{3}{4} * (3/7)^3 * 1/14 * 1/14 \\ \approx 0.0003$$

Conditional Probabilities:

$$P(\text{Chinese}|c) = (5+1) / (8+6) = 6/14 = 3/7$$

$$P(\text{Tokyo}|c) = (0+1) / (8+6) = 1/14$$

$$P(\text{Japan}|c) = (0+1) / (8+6) = 1/14$$

$$P(\text{Chinese}|j) = (1+1) / (3+6) = 2/9$$

$$P(\text{Tokyo}|j) = (1+1) / (3+6) = 2/9$$

$$P(\text{Japan}|j) = (1+1) / (3+6) = 2/9$$

$$P(j|d5) \propto \frac{1}{4} * (2/9)^3 * 2/9 * 2/9 \\ \approx 0.0001$$

37

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'

Summary: Naive Bayes is Not So Naive

- Very fast, low storage requirements
- Robust to irrelevant features
- Very good in domains with many equally important features
- Optimal if the independence assumptions hold
- A good dependable baseline for text classification
 - **But often other classifiers give better accuracy**