Text Classification and Naïve Bayes

Dr. Demetrios Glinos

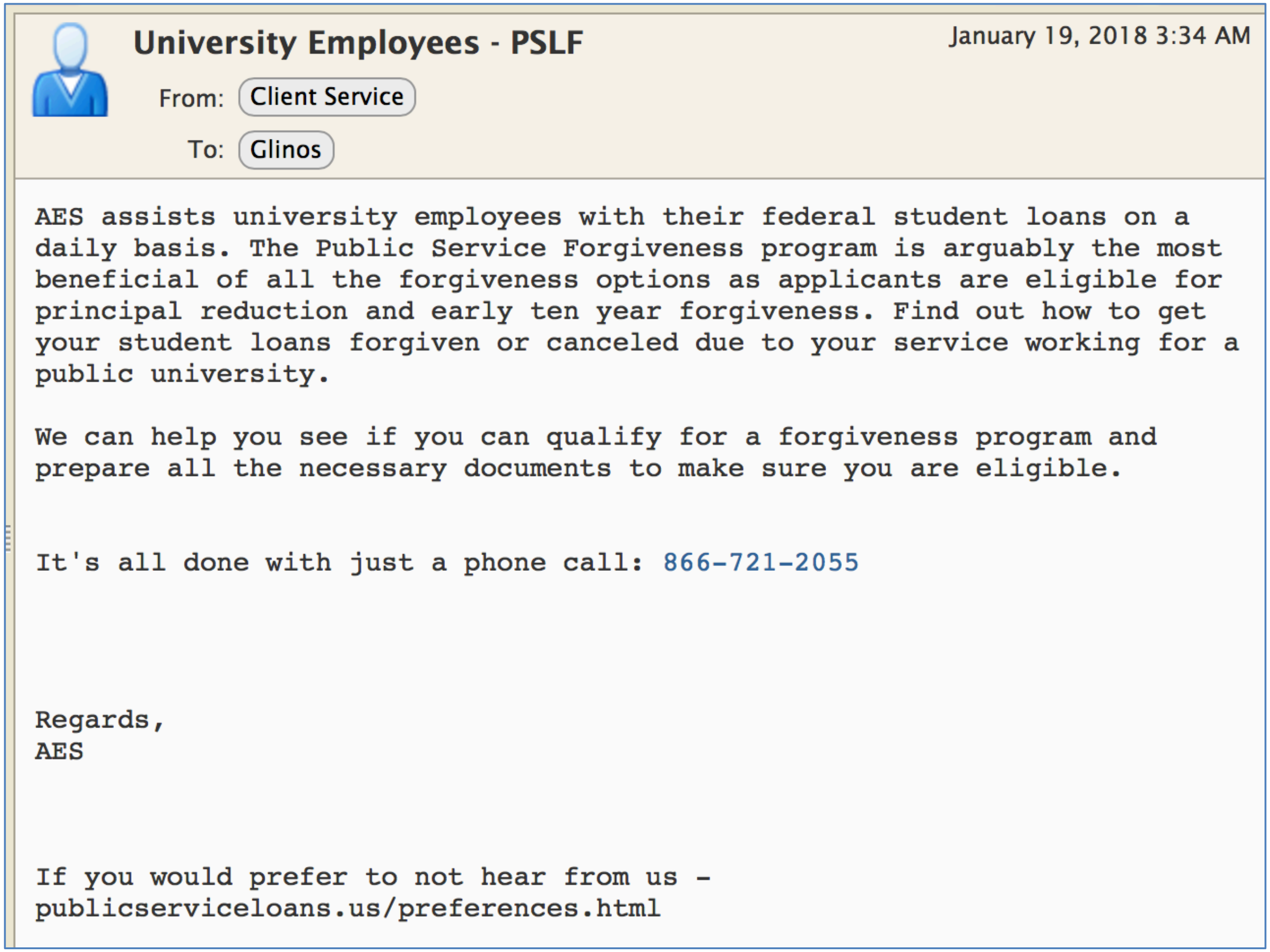
University of Central Florida

CAP6640 – Computer Understanding of Natural Language

Today

- The Text Classification Task
- Naïve Bayes Classification
- Learning a Naïve Bayes Classifier
- Relationship to Language Modeling
- Multinomial Naïve Bayes Example
- Precision, Recall, and the F measure
- Handling More than Two Classes
- Practical Issues in Text Classification

Is This Spam?



University Employees - PSLF

January 19, 2018 3:34 AM

From: Client Service

To: Glinos

AES assists university employees with their federal student loans on a daily basis. The Public Service Forgiveness program is arguably the most beneficial of all the forgiveness options as applicants are eligible for principal reduction and early ten year forgiveness. Find out how to get your student loans forgiven or canceled due to your service working for a public university.

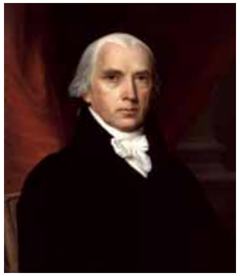
We can help you see if you can qualify for a forgiveness program and prepare all the necessary documents to make sure you are eligible.

It's all done with just a phone call: 866-721-2055

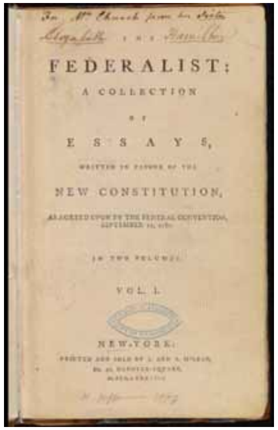
Regards,
AES

If you would prefer to not hear from us - publicserviceloans.us/preferences.html

Who Wrote the Disputed Federalist Papers?



James Madison



For Mrs Church from her Sister

Elizabeth THE Hamilton

FEDERALIST:

A COLLECTION

OF

ESSAYS,

WRITTEN IN FAVOUR OF THE

NEW CONSTITUTION,

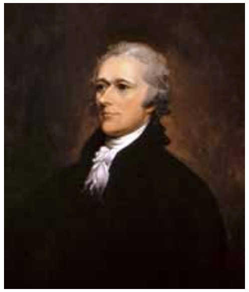
AS AGREED UPON BY THE FEDERAL CONVENTION,
SEPTEMBER 17, 1787.

IN TWO VOLUMES.

VOL. I.

NEW-YORK:

PRINTED AND SOLD BY J. AND A. M'LEAN,
No. 41, HANOVER-SQUARE,
M,DCC,LXXXVIII.

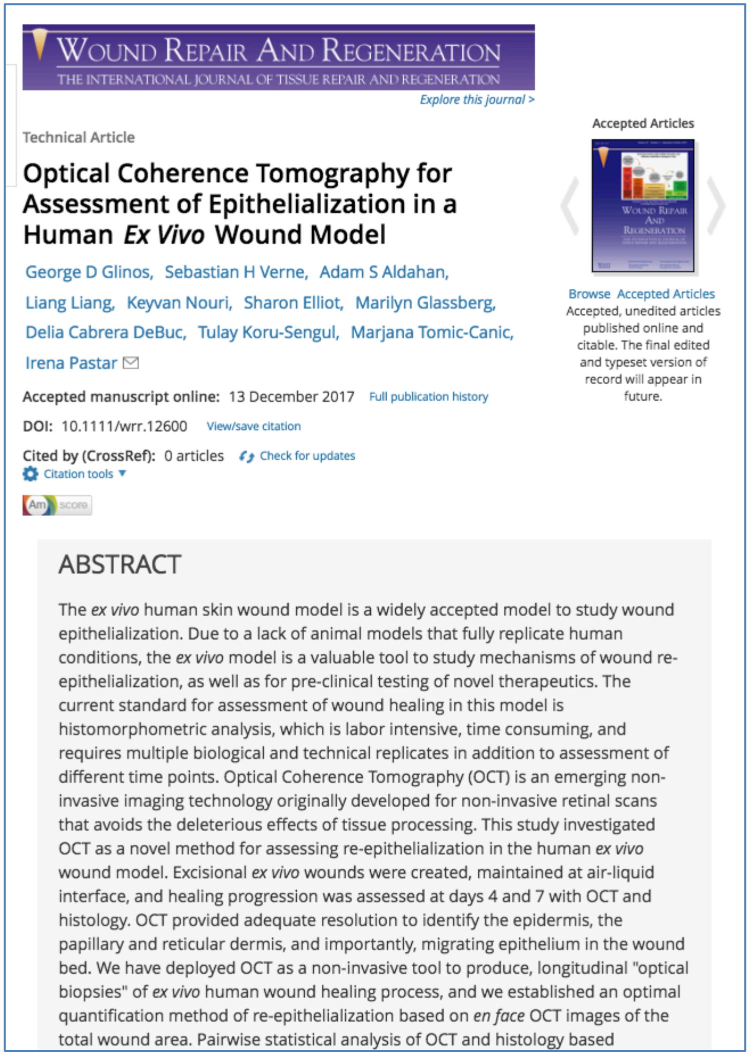


Alexander Hamilton

- Federalist papers
  - 85 anonymous essays published in 1787 - 1788 to convince New York to ratify the U.S. Constitution
  - Authors: Alexander Hamilton, James Madison, John Jay
  - Authorship of 12 papers disputed after Alexander Hamilton's death

- Mosteller and Wallace (1962)
  - Resolved authorship of disputed papers using Bayesian methods
    - All 12 were authored by Madison
  - Final tally: Hamilton (51), Madison (29), Jay (5)

Topic Extraction



WOUND REPAIR AND REGENERATION
THE INTERNATIONAL JOURNAL OF TISSUE REPAIR AND REGENERATION
Explore this journal >

Technical Article

Optical Coherence Tomography for Assessment of Epithelialization in a Human *Ex Vivo* Wound Model

George D Glinos, Sebastian H Verne, Adam S Aldahan, Liang Liang, Keyvan Nouri, Sharon Elliot, Marilyn Glassberg, Delia Cabrera DeBuc, Tulay Koru-Sengul, Marjana Tomic-Canic, Irena Pastar ✉

Accepted manuscript online: 13 December 2017 Full publication history

DOI: 10.1111/wrr.12600 View/save citation

Cited by (CrossRef): 0 articles Check for updates

Citation tools ▾

Accepted Articles

Browse Accepted Articles

Accepted, unedited articles published online and citable. The final edited and typeset version of record will appear in future.

ABSTRACT

The *ex vivo* human skin wound model is a widely accepted model to study wound epithelialization. Due to a lack of animal models that fully replicate human conditions, the *ex vivo* model is a valuable tool to study mechanisms of wound re-epithelialization, as well as for pre-clinical testing of novel therapeutics. The current standard for assessment of wound healing in this model is histomorphometric analysis, which is labor intensive, time consuming, and requires multiple biological and technical replicates in addition to assessment of different time points. Optical Coherence Tomography (OCT) is an emerging non-invasive imaging technology originally developed for non-invasive retinal scans that avoids the deleterious effects of tissue processing. This study investigated OCT as a novel method for assessing re-epithelialization in the human *ex vivo* wound model. Excisional *ex vivo* wounds were created, maintained at air-liquid interface, and healing progression was assessed at days 4 and 7 with OCT and histology. OCT provided adequate resolution to identify the epidermis, the papillary and reticular dermis, and importantly, migrating epithelium in the wound bed. We have deployed OCT as a non-invasive tool to produce, longitudinal "optical biopsies" of *ex vivo* human wound healing process, and we established an optimal quantification method of re-epithelialization based on *en face* OCT images of the total wound area. Pairwise statistical analysis of OCT and histology based

- Topic categories
  - Epithelialization
  - Histology
  - Human Wound Healing
  - Optical Coherence Tomography

Text Classification Research Areas

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

Text Classification Defined

- Input:
  - a document **d**
    - anything textual: sentence, paragraph, complete document, even a computer program
  - a fixed set of classes $\mathbf{C = \{ c_1, c_2, ..., c_n \}}$
- Output:
  - a predicted class $\mathbf{c} \in \mathbf{C}$

Text Classification: Hand-Coded Rules

- Develop rules based on attributes (features)
- Example, for a spam filter, features could be
  - words: e.g., "FREE"
  - text patterns: $nnn, all upper case
  - non-text information: is sender in the recipient's contact list, etc.
- rules crafted by domain experts
- accuracy can be high
- developing and maintaining rule sets can be expensive

> Dear Sir.
>
> First, I must solicit your confidence in this transaction, this is by virtue of its nature as being utterly confidencial and top secret. ...

> TO BE REMOVED FROM FUTURE MAILINGS, SIMPLY REPLY TO THIS MESSAGE AND PUT "REMOVE" IN THE SUBJECT.
>
> 99 MILLION EMAIL ADDRESSES FOR ONLY $99

> Ok, Iknow this is blatantly OT but I'm beginning to go insane. Had an old Dell Dimension XPS sitting in the corner and decided to put it to use, I know it was working pre being stuck in the corner, but when I plugged it in, hit the power nothing happened.

Text Classification: Data Compression

- Lossless compression algorithms
  - input: sequence of symbols
  - action: find repeated patterns in sequence
  - output: a more compact description of the sequence

- Data compression approach
  - can use off-the-shelf algorithms
    - LZW, gzip, RAR
    - good accuracy, but can be slow
  - lump together spam and compress
  - lumg together ham and compress
  - for new message:
    - lump together with spam/ham and compress
    - whichever class compresses better is the winner

Text Classification: Language Modeling

- Example data for a spam filter
  - emails in spam folder
  - emails in inbox

e.g., word bigrams: “for cheap”
e.g., character bigrams: FR RE EE

- Language-modeling approach
  - develop n-gram model for P(Message|spam) based on spam folder
  - develop n-gram model for P(Message|ham) based on inbox folder
  - classify new message using Bayes’ rule:

$$\underset{c\in\{spam,ham\}}{\text{argmax}}\ P(c\,|\,message) = \underset{c\in\{spam,ham\}}{\text{argmax}}\ P(message\,|\,c)P(c)$$

  - where $P(c = spam)$ is the relative proportion of the spam corpus
  - and where we apply the Markov assumption to obtain (assuming bigrams)

$$P(message|c) = \prod_{i=1}^{N} [\,P(t_i|t_{i-1}, c)]$$

Text Classification: Machine Learning

- Basic idea:
  - represent data (messages) as feature/value pairs
  - features:
    - words (unigrams) in the vocabulary (can be very large, e.g., 100K)
      - unigram representation also called “bag of words” model
    - can also be n-grams
    - can include non-n-gram values: time sent, whether includes image, etc.
  - values: number of times each word appears in the message
  - most features will have zero values

- Feature selection
  - may use a reduced set of features (e.g., top 100) to simplify complexity

- Machine learning algorithms used:
  - multiclass perceptrons, NN, SVM, K-NN, decision trees, NaïveBayes

Today

- The Text Classification Task
- Naïve Bayes Classification
- Learning a Naïve Bayes Classifier
- Relationship to Language Modeling
- Multinomial Naïve Bayes Example
- Precision, Recall, and the F measure
- Handling More than Two Classes
- Practical Issues in Text Classification

Naïve Bayes Intuition

- A type of supervised learning
    - classifier is learned from a training corpus of hand-labeled documents

- Simple ("naïve") classification method based on Bayes rule:

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}$$

- Relies on representing a document as a **bag of words**
    - a very simple representation
    - word order does not matter
    - word count does matter

Example: Bag-of-words representation

- Consider the sentence

"how much wood would a woodchuck chuck if a woodchuck could chuck wood"

- Represented as a bag of words →

word	count
a	2
chuck	2
could	1
how	1
if	1
much	1
wood	2
woodchuck	2
would	1

- classification involves mapping this vector of features to a class

Statistical Background

- Bayes' Rule

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

- Chain Rule for probabilities

$$P(x_1,\ldots,x_n) = \prod_i P(x_i \mid x_1,\ldots,x_{i-1})$$
$$= P(x_1)\,P(x_2\,|\,x_1\,)\,P(x_3\,|\,x_1\,,x_2\,)\,P(x_4\,|\,x_1\,,x_2\,,x_3)\cdots$$

Most Likely Class

- We use the **MAP ("maximum a posteriori")** determination of the most likely class

$$c_{MAP} = argmax_{c \in C}\, P(c|d)$$

- Applying Bayes' Rule

$$c_{MAP} = argmax_{c \in C} \frac{P(d|c)P(c)}{P(d)}$$

- And simply dropping the denominator

$$c_{MAP} = argmax_{c \in C} P(d|c)P(c)$$

- Representing the document as a *feature vector*

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

Naïve Bayes Parameter Space

$$c_{MAP} = argmax_{c \in C} P(\underbrace{x_1, x_2, \ldots, x_n}_{\textit{feature vector}}|c)P(c)$$

- Suppose each x_i represents a word in a particular position in the document
  - e.g. x_{35} = "woodchuck"

- Suppose also that there are |X| possible words in each position and a maximum of n word positions in any document

- Then, there are $O(\ |X|^n \bullet |C|\ )$ possible input documents, hence probabilities, that we would need to calculate

- We would need a really large set of training examples to estimate all these probabilities, since the test set can include any of them

Multinomial NB Simplifying Assumptions

- Bag-of-words assumption
  - assume position doesn't matter
  - but *multiplicity* does
- Conditional independence
  - assume feature probabilities P(x_i | c_j) are conditionally independent given the class c

$$P(x_1, x_2, \dots, x_n | c) = \prod_{i=1}^{n} P(\, x_i \,| c\,)$$

- As a result, we need to compute only |X|•C probabilities

Multinomial Naïve Bayes Classifier

- From previous slide: $P(x_1, x_2, \dots, x_n|c) = \prod_{i=1}^{n} P(\,x_i\,|c\,)$
- Recall from before $c_{MAP} \;= argmax_{c\in C} P(x_1, x_2, \dots, x_n|c)P(c)$
- Therefore

$$c_{NB} \;= argmax_{c_j\in C}\; P(c_j) \prod_{i=1}^{n} P(\,x_i\,|\,c_j\,)$$

where i ranges over all word positions in the document

Today

- The Text Classification Task
- Naïve Bayes Classification
- Learning a Naïve Bayes Classifier
- Relationship to Language Modeling
- Multinomial Naïve Bayes Example
- Precision, Recall, and the F measure
- Handling More than Two Classes
- Practical Issues in Text Classification

Learning a Multinomial NB Model

- **Maximum likelihood estimate (MLE)**
  - simply use the frequencies observed in the data

$$\hat{P}(\,c_j\,) = \frac{|\,\{d \in docs \,:\, class(\,d\,) = \,c_j\}\,|}{|\,\{docs\}\,|}$$

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

- Create a mega-document for topic (class) j by concatenating all documents in this topic
  - use the frequency of w in mega-document

The Problem of Zeroes

- For any particular word w_k that did not occur in the training data for class c_j

$$\hat{P}(\, w_k \mid c_j) = \frac{count(w_k, c_j\,)}{\sum_{w \in V} count(\, w, c_j\,)} = 0$$

- But w_k *could* appear in the test data for class c_j

- We can use Laplace (add-1) smoothing to ensure non-zero probability for words like w_k

$$\hat{P}(\, w_k \mid c_j) = \frac{count(w_k, c_j\,) + 1}{\sum_{w \in V}(count(\, w, c_j\,) + 1)}$$

$$= \frac{count(w_k, c_j\,) + 1}{(\sum_{w \in V} count(\, w, c_j\,)) + |V|}$$

Multinomial Naïve Bayes Learning

- From the training corpus, extract the *Vocabulary*

- Calculate P(c_j) terms

for each c_j in C, find docs$_j$ = { all docs with class = c_j }

$$P(\, c_j \,) \;=\; \frac{|\, docsj \,|}{|\, total \;\#\; documents \,|}$$

- Calculate P(w_k | c_j) terms

Let *Text$_j$* = a mega-doc containing all docs in doc$_j$

for each word w_k in *Vocabulary*, compute

n_k = # occurrences of w_k in *Test$_j$*

$$P(\, wk \mid c_j \,) = \frac{n_k + \ \alpha}{n + \alpha|Vocabulary|}$$

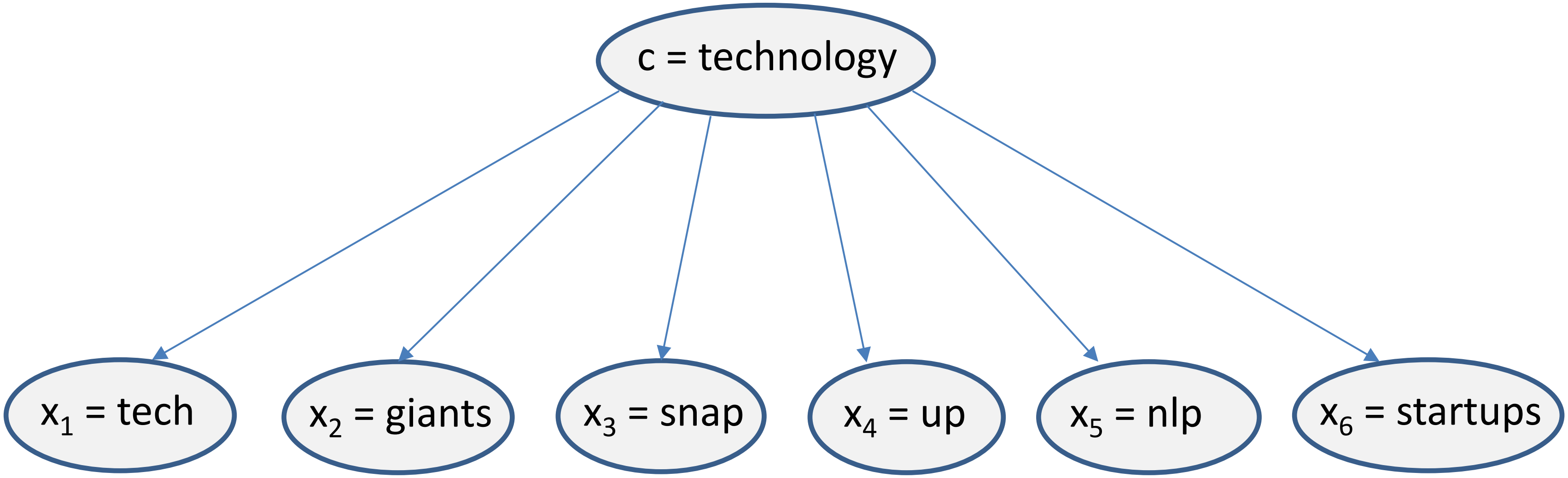
where α = 1 for Laplace

Today

- The Text Classification Task
- Naïve Bayes Classification
- Learning a Naïve Bayes Classifier
- Relationship to Language Modeling
- Multinomial Naïve Bayes Example
- Precision, Recall, and the F measure
- Handling More than Two Classes
- Practical Issues in Text Classification

Generative Model for Multinomial NB

- The conditional probabilities we calculate may be considered a generative model for the words in the vocabulary for each possible topic (class)



Naïve Bayes and Language Modeling

- Naïve Bayes classifiers can use any features for classifications
  - e.g., words, phrases, URLs, email addresses, network features
- But if,
  - we use only individual words as features
  - and we use all of the words in the text, not just a subset
- Then
  - Naïve Bayes has an important similarity to language modeling:
    - Each class is a unigram language model
    - Compute for each word: $P(\,word \mid c\,)$
    - Assign for each sentence: $P(\,s \mid c\,) = \prod P(\,word \mid c\,)$

Naïve Bayes as a Language Model

- Which class assigns the higher probability to s = "tech giants snap up nlp startups" ?

c = finance

w_i	P(w_i | c)
tech	.1
giants	.05
snap	.02
up	.3
nlp	.03
startups	.3

c = sports

w_i	P(w_i | c)
tech	.01
giants	.4
snap	.3
up	.05
nlp	.0001
startups	.01

class	tech	giants	snap	up	nlp	startups	P(s | c))
finance	.1	.05	.02	.3	.03	.3	2.7×10^{-7}
sports	.01	.4	.3	.05	.0001	.01	6×10^{-11}

Today

- The Text Classification Task
- Naïve Bayes Classification
- Learning a Naïve Bayes Classifier
- Relationship to Language Modeling
- Multinomial Naïve Bayes Example
- Precision, Recall, and the F measure
- Handling More than Two Classes
- Practical Issues in Text Classification

Multinomial Naïve Bayes Example

- Consider a small database of movie reviews

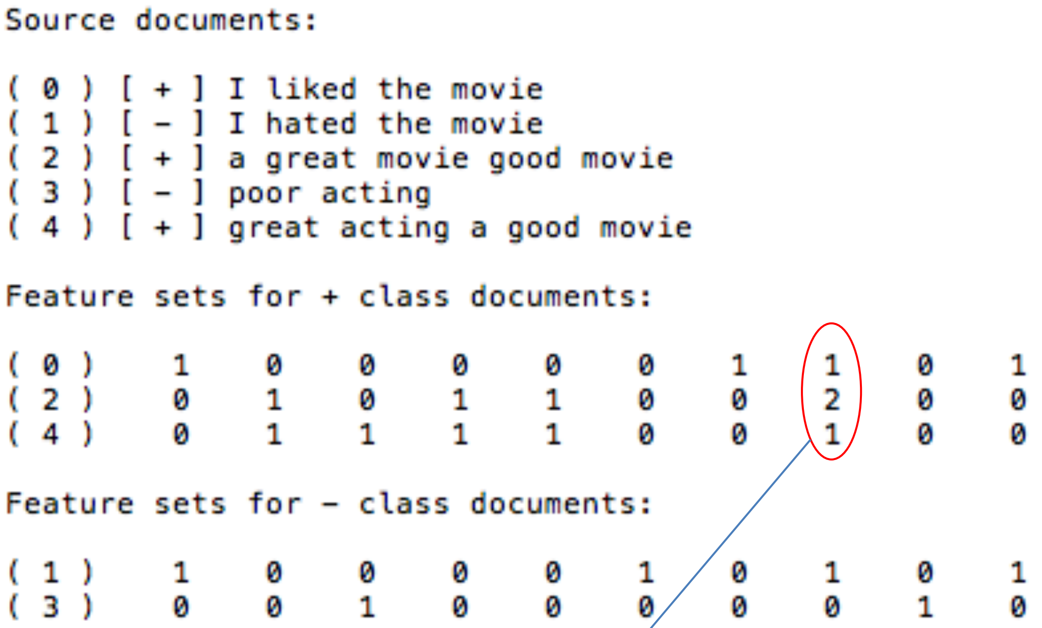
"I liked the movie"	(+)
"I hated the movie"	(-)
"a great movie a good movie"	(+)
"poor acting"	(-)
"great acting a good movie"	(+)

- Computing the priors

P(+) = 3 / 5 = .6

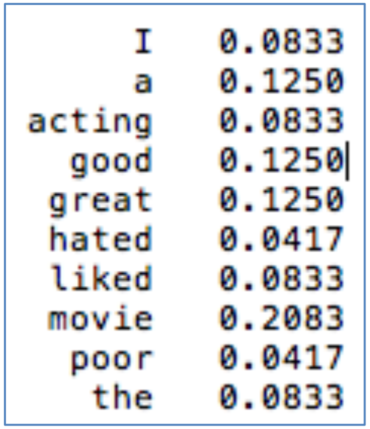
P(-) = 2 / 5 = .4

Multinomial Naïve Bayes Example



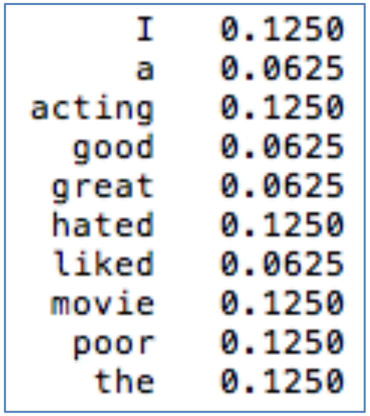
```
Source documents:

( 0 ) [ + ] I liked the movie
( 1 ) [ - ] I hated the movie
( 2 ) [ + ] a great movie good movie
( 3 ) [ - ] poor acting
( 4 ) [ + ] great acting a good movie

Feature sets for + class documents:

( 0 )    1    0    0    0    0    0    1    1    0    1
( 2 )    0    1    0    1    1    0    0    2    0    0
( 4 )    0    1    1    1    1    0    0    1    0    0

Feature sets for - class documents:

( 1 )    1    0    0    0    0    1    0    1    0    1
( 3 )    0    0    1    0    0    0    0    0    1    0
```

$P(x_i \mid +)$



```
     I   0.0833
     a   0.1250
acting   0.0833
  good   0.1250
 great   0.1250
 hated   0.0417
 liked   0.0833
 movie   0.2083
  poor   0.0417
   the   0.0833
```

$P(x_i \mid -)$



```
     I   0.1250
     a   0.0625
acting   0.1250
  good   0.0625
 great   0.0625
 hated   0.1250
 liked   0.0625
 movie   0.1250
  poor   0.1250
   the   0.1250
```

Example: P(movie | +) = (4+1) / (14+10) = .2083

Multinomial Naïve Bayes Example

Test sentence: s = "I hated the poor acting"

P(+ | s) = (.6)(.0833)(.0417)(.0833)(.0417)(.0833)

= 6.028 × 10^{-7}

P(- | s) = (.4)(.125)(.125)(.125)(.125)(.125)

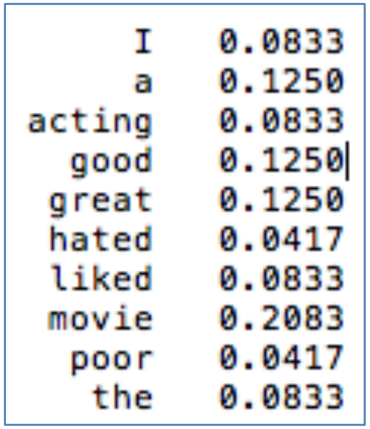
= 1.221 × 10^{-5}

Using sums of log values:

P(+ | s) = - 14.32

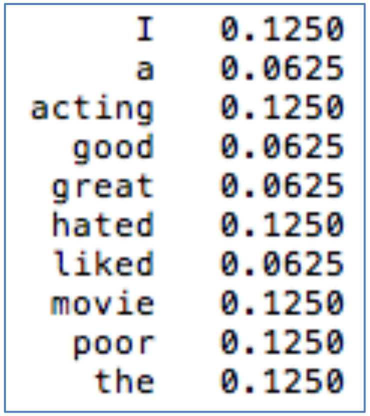
P(- | s) = - 11.31

P(x_i | +)



```
      I    0.0833
      a    0.1250
 acting    0.0833
   good    0.1250
  great    0.1250
  hated    0.0417
  liked    0.0833
  movie    0.2083
   poor    0.0417
    the    0.0833
```

P(x_i | -)



```
      I    0.1250
      a    0.0625
 acting    0.1250
   good    0.0625
  great    0.0625
  hated    0.1250
  liked    0.0625
  movie    0.1250
   poor    0.1250
    the    0.1250
```

Naïve Bayes Performance

- Very fast, low storage requirements
- Robust to irrelevant features
    - irrelevant features cancel out each other without affecting results
- Performs well in domains with many equally important features
- Optimal if the independence assumption holds
- Overall, a good dependable baseline for text classification
    - but there are other classifiers that give better accuracy

Today

- The Text Classification Task
- Naïve Bayes Classification
- Learning a Naïve Bayes Classifier
- Relationship to Language Modeling
- Multinomial Naïve Bayes Example
- Precision, Recall, and the F measure
- Handling More than Two Classes
- Practical Issues in Text Classification

System Testing Outcomes

- Consider a binary classifier
- It can "select" as a positive instance
  - a document that should be selected ← tp ("true positive")
  - a document that should not be selected ← fp ("false positive")
- It can fail to select as a positive instance
  - a document that should not be selected ← tn ("true negative")
  - a document that should be selected ← fn ("false negative")
- 2-by-2 contingency table

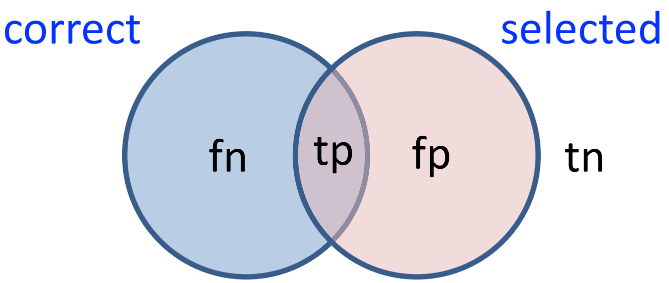
	should select	should not select
is selected	tp	fp
is not selected	fn	tn

Precision and Recall

	should select	should not select
is selected	tp	fp
is not selected	fn	tn

- **Precision**
  - % of selected items that are correct = tp / (tp + fp)

- **Recall**
  - % of correct items that are selected = tp / (tp + fn)



The F measure

- Precision and recall are competing measures
  - a system can have high precision, but very low recall
  - a system can have high recall, but very low precision

- **F measure**
  - a measure that assessed the tradeoff between precision and recall
  - is the weighted harmonic mean

$$F_\beta = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R}$$

- **F$_1$ score** is typically used (i. e., β = 1, so α = ½) : $F_1 = \frac{2PR}{P+R}$

Today

- The Text Classification Task
- Naïve Bayes Classification
- Learning a Naïve Bayes Classifier
- Relationship to Language Modeling
- Multinomial Naïve Bayes Example
- Precision, Recall, and the F measure
- Handling More than Two Classes
- Practical Issues in Text Classification

Multivalue Classification

- **Multivalue ("any-of") classification**
  - document can belong to 0, 1, or >1 classes

- Use multiple binary classifiers
  - for each class c ∈ C
  - build a classifier γ_c to distinguish c from other classes c' ∈ C
  - given a test document d
    - evaluate it for membership in each class c using γ_c
    - d belongs to ***every*** class for which γ_c returns true

Multinomial Classification

- **Multinomial ("one-of") classification**
  - classes are mutually exclusive
  - document belongs to exactly one class

- Use multiple binary classifiers
  - for each class c ∈ C
  - build a classifier γ_c to distinguish c from other classes c' ∈ C
  - given a test document d
    - evaluate it for membership in each class c using γ_c
    - d belongs to the class for which γ_c returns the ***maximum score***

Confusion matrix c

- When there are several classes, a member of one class may be incorrectly categorized as a member of another class

	Assigned politics	Assigned finance	Assigned sports	Assigned entertainment
True politics	tp			
True finance		tp		
True sports			tp	
True entertainment				tp

- Entries along main diagonal are true positives
- Entries in other cells are false positive for assigned class, and false negatives for true class

Per class evaluation measures

- Recall
  - fraction of docs in class i classified correctly

$$= \frac{c_{ii}}{\sum_j c_{ij}}$$

- Precision
  - fraction of docs assigned class i that should be class i

$$= \frac{c_{ii}}{\sum_j c_{ji}}$$

- Accuracy
  - fraction of docs that are classified correctly
  - = 1 – error rate

$$= \frac{\sum_i c_{ii}}{\sum_j \sum_i c_{ij}}$$

Micro- vs. Macro-Averaging

- How to combine performace from all classes, or to combine results from multiple splits when cross-validating
  - Macro-averaging
    - compute performance for each class, then average
    - i.e., average all the precision values to get overall precision
  - Micro-averaging
    - include data from all classes to compute overall performace
    - i.e., add all true positives and false positives to compute global recall

Example: Combined Precision

- Consider these contingency tables:

Class 1

	Truth : yes	Truth : no
Classifier: yes	10	10
Classifier: no	10	970

Class 2

	Truth : yes	Truth : no
Classifier: yes	90	10
Classifier: no	10	890

Micro-average

	Truth : yes	Truth : no
Classifier: yes	100	20
Classifier: no	20	1860

- Macro-averaged precision: (0.5 + 0.9) / 2 = 0.7

- Micro-averaged precision: 100 /120 = 0.83

Today

- The Text Classification Task
- Naïve Bayes Classification
- Learning a Naïve Bayes Classifier
- Relationship to Language Modeling
- Multinomial Naïve Bayes Example
- Precision, Recall, and the F measure
- Handling More than Two Classes
- Practical Issues in Text Classification

Practical Issue: Underflow Prevention

- Multiplying many probabilities can result in floating-point underflow
- Better to sum logs of probabilities instead of multiplying the probabilities
  - since log(xy) = log(x) + log(y)
- The class with the highest un-normalized log probability score is still the most probable

$$c_{NB} = argmax_{c_j \in C} \left[\log P(c_j) + \sum_i \log P(x_i \mid c_j) \right]$$

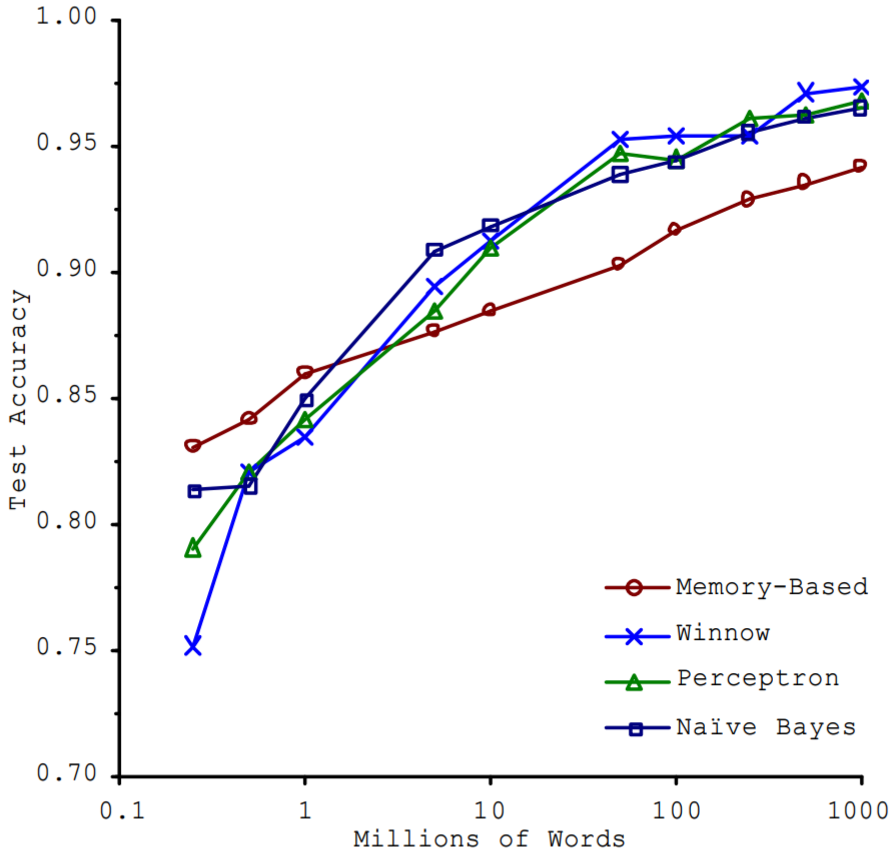
- Note: these values will all be negative

Practical Issue: Data Availability

- The type of text classification we can do often depends on the available data
  - No training data
    - generally use hand-coded rules
    - need careful crafting, generally very time-consuming
  - Very little training data
    - use Naïve Bayes
    - try semi-supervised methods – e.g., bootstrapping
  - A reasonable amount of data
    - SVMs, Logistic regression, decision trees
  - Huge amount of data (highest accuracy)
    - SVMs (train time) and k-NN (test time) can be too slow
    - Naïve Bayes can be useful here

Accuracy as a Function of Data Size

- With enough data, the classifier may not matter



source:

Banko and Brill (2001)

paper on WSD