



# Text Classification and Naïve Bayes

## The Task of Text Classification

Dan Jurafsky



# Is this spam?

**Subject: Important notice!**

**From:** Stanford University <newsforum@stanford.edu>

**Date:** October 28, 2011 12:34:16 PM PDT

**To:** undisclosed-recipients;;

---

Greats News!

You can now access the latest news by using the link below to login to Stanford University News Forum.

<http://www.123contactform.com/contact-form-StanfordNew1-236335.html>

Click on the above link to login for more information about this new exciting forum. You can also copy the above link to your browser bar and login for more information about the new services.

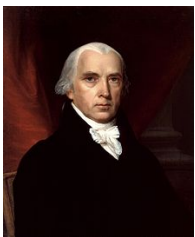
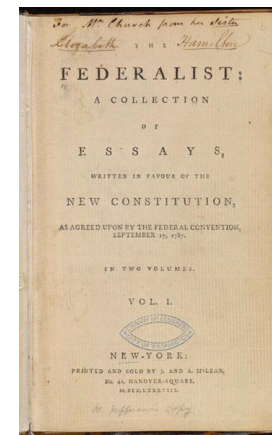
© Stanford University. All Rights Reserved.

Dan Jurafsky

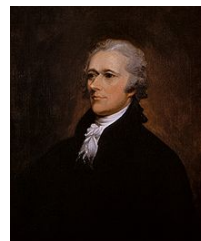


# Who wrote which Federalist papers?

- 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



## Male or female author?

1. By 1925 present-day Vietnam was divided into three parts under French colonial rule. The southern region embracing Saigon and the Mekong delta was the colony of Cochin-China; the central area with its imperial capital at Hue was the protectorate of Annam...
2. Clara never failed to be astonished by the extraordinary felicity of her own name. She found it hard to trust herself to the mercy of fate, which had managed over the years to convert her greatest shame into one of her greatest assets...



## 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

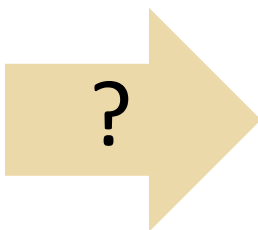


- It was pathetic. The worst part about it was the boxing scenes.



# What is the subject of this article?

## MEDLINE Article



## MeSH Subject Category Hierarchy

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

Dan Jurafsky



# 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  $c \in C$



Dan Jurafsky



# 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  $\gamma: d \rightarrow c$

Dan Jurafsky



# Classification Methods: Supervised Machine Learning

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





Dan Jurafsky



## 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

Y (

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.

) = C





## The bag of words representation

Y (

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.

) = C







# The bag of words representation: using a subset of words

Y (

```
x love xxxxxxxxxxxxxxxxxxxx sweet
xxxxxxxx satirical xxxxxxxxxxxx
xxxxxxxxxxxx great xxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxx fun xxxx
xxxxxxxxxxxxxxxxxxxx whimsical xxxx
romantic xxxx laughing
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxx recommend xxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx several xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxx happy xxxxxxxxxxxx again
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

) = C





# The bag of words representation

$Y$  (

great	2
love	2
recommend	1
laugh	1
happy	1
...	...

)

=  $C$





# Bag of words for document classification

Test document

parser  
language  
label  
translation  
...

?

Machine Learning	NLP	Garbage Collection	Planning	GUI
learning	<u>parser</u>	garbage	planning	...
<u>training</u>	tag	collection	temporal	
algorithm	training	memory	reasoning	
shrinkage	<u>translation</u>	optimization	plan	
network...	<u>language...</u>	region...	<u>language...</u>	





Dan Jurafsky



# Bayes' Rule Applied to Documents and Classes

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

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



## Naïve Bayes Classifier (I)

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

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



## Naïve Bayes Classifier (II)

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

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

Document  $d$   
represented as  
features  
 $x_1 \dots x_n$





## Naïve Bayes Classifier (IV)

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

$O(|X|^n \cdot |C|)$  parameters

Could only be estimated if a very, very large number of training examples was available.

How often does this class occur?

We can just count the relative frequencies in a corpus



# Multinomial Naïve 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  $c$ .

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

Dan Jurafsky



## Multinomial Naïve Bayes Classifier

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

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

Dan Jurafsky



# Applying Multinomial Naive Bayes Classifiers to Text Classification

positions  $\leftarrow$  all word positions in test document

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



# Text Classification and Naïve Bayes

## Formalizing the Naïve Bayes Classifier



# Text Classification and Naïve Bayes

Naïve Bayes:  
Learning



## Learning the Multinomial Naïve Bayes Model

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

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

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



## 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





## 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!

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



## Laplace (add-1) smoothing for Naïve Bayes

$$\begin{aligned}\hat{P}(w_i | c) &= \frac{\mathit{count}(w_i, c) + 1}{\sum_{w \in V} (\mathit{count}(w, c) + 1)} \\ &= \frac{\mathit{count}(w_i, c) + 1}{\left( \sum_{w \in V} \mathit{count}(w, c) \right) + |V|}\end{aligned}$$



# 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  $P(w_k | c_j)$  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  $Text_j$
$$P(w_k | c_j) \leftarrow \frac{n_k + \alpha}{n + \alpha |Vocabulary|}$$



## Laplace (add-1) smoothing: unknown words

Add one extra word to the vocabulary, the “unknown word”  $w_u$

$$\begin{aligned}\hat{P}(w_u | c) &= \frac{\text{count}(w_u, c) + 1}{\left( \sum_{w \in V} \text{count}(w, c) \right) + |V + 1|} \\ &= \frac{1}{\left( \sum_{w \in V} \text{count}(w, c) \right) + |V + 1|}\end{aligned}$$



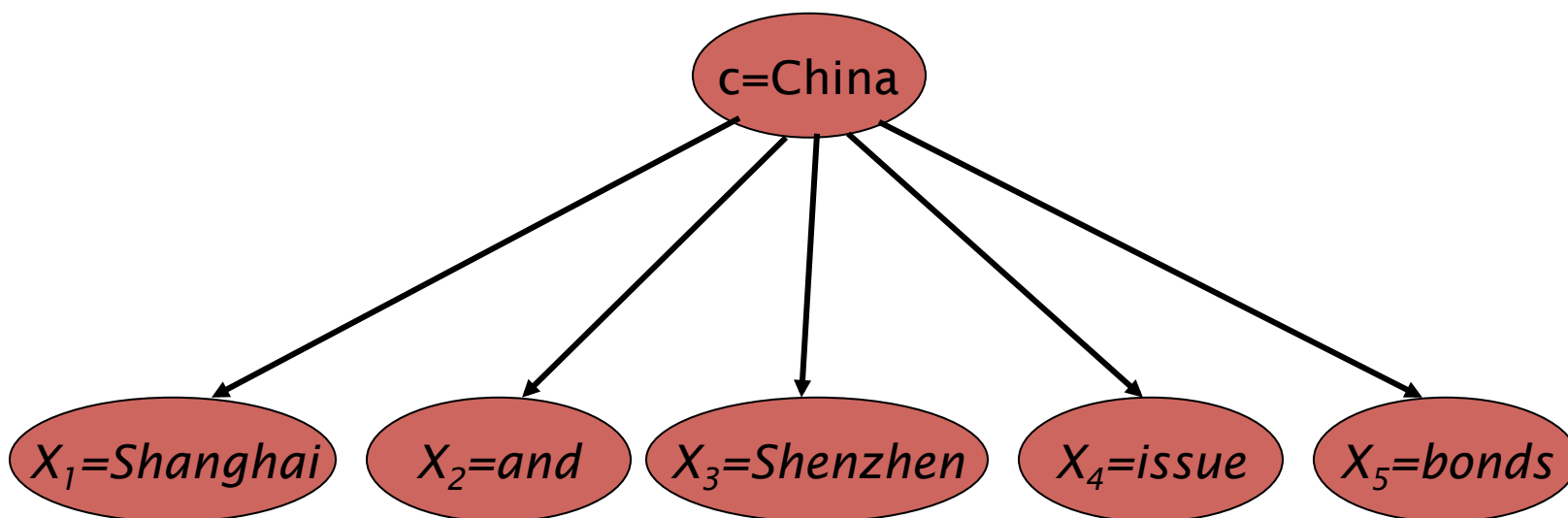
# Text Classification and Naïve Bayes

Naïve Bayes:  
Learning





## Generative Model for Multinomial Naïve Bayes





## Naïve Bayes and Language Modeling

- Naïve 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
  - Naïve bayes has an important similarity to language modeling.





## 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	<u>I</u>	<u>love</u>	<u>this</u>	<u>fun</u>	<u>film</u>
0.1	love	0.1	0.1	.05	0.01 0.1
0.01	this				
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  $s$ ?

Model pos	
0.1	I
0.1	love
0.01	this
0.05	fun
0.1	film

Model neg	
0.2	I
0.001	love
0.01	this
0.005	fun
0.1	film

<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

$$P(s|\text{pos}) > P(s|\text{neg})$$



# Text Classification and Naïve Bayes

Naïve Bayes:  
Relationship to  
Language Modeling





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

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

	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:**

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

$$P(j) = \frac{1}{4}$$

**Conditional Probabilities:**

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

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

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

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

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

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

**Choosing a class:**

$$P(c | d5) \propto 3/4 * (3/7)^3 * 1/14 * 1/14$$

$$\approx 0.0003$$

$$P(j | d5) \propto 1/4 * (2/9)^3 * 2/9 * 2/9$$

$$\approx 0.0001$$

Dan Jurafsky



# 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)



## 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**



# Text Classification and Naïve Bayes

Multinomial Naïve  
Bayes: A Worked  
Example





Dan Jurafsky



## The 2-by-2 contingency table

	correct	not correct
selected	tp	fp
not selected	fn	tn



## Precision and recall

- **Precision:** % of selected items that are correct  
**Recall:** % of correct items that are selected

	correct	not correct
selected	tp	fp
not selected	fn	tn



## A combined measure: F

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

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

- The harmonic mean is a very conservative average; see *IIR* § 8.3
- People usually use balanced F1 measure
  - i.e., with  $\beta = 1$  (that is,  $\alpha = \frac{1}{2}$ ):  $F = 2PR/(P+R)$







## More Than Two Classes: Sets of binary classifiers

- Dealing with **any-of** or **multivalued** classification
  - A document can belong to 0, 1, or >1 classes.
- For each class  $c \in C$ 
  - Build a classifier  $\gamma_c$  to distinguish  $c$  from all other classes  $c' \in C$
- Given test doc  $d$ ,
  - Evaluate it for membership in each class using each  $\gamma_c$
  - $d$  belongs to **any** class for which  $\gamma_c$  returns true



## More Than Two Classes: Sets of binary classifiers

- One-of or multinomial classification
  - Classes are mutually exclusive: each document in exactly one class
- For each class  $c \in C$ 
  - Build a classifier  $\gamma_c$  to distinguish  $c$  from all other classes  $c' \in C$
- Given test doc  $d$ ,
  - Evaluate it for membership in each class using each  $\gamma_c$
  - $d$  belongs to the one class with maximum score





# Evaluation: Classic Reuters-21578 Data Set

- Most (over)used data set, 21,578 docs (each 90 types, 200 tokens)
- 9603 training, 3299 test articles (ModApte/Lewis split)
- 118 categories
  - An article can be in more than one category
  - Learn 118 binary category distinctions
- Average document (with at least one category) has 1.24 classes
- Only about 10 out of 118 categories are large

Common categories  
(#train, #test)

57

- |                            |                       |
|----------------------------|-----------------------|
| • Earn (2877, 1087)        | • Trade (369,119)     |
| • Acquisitions (1650, 179) | • Interest (347, 131) |
| • Money-fx (538, 179)      | • Ship (197, 89)      |
| • Grain (433, 149)         | • Wheat (212, 71)     |
| • Crude (389, 189)         | • Corn (182, 56)      |



# Reuters Text Categorization data set (Reuters-21578) document

<REUTERS TOPICS="YES" LEWISSPLIT="TRAIN" CGISPLIT="TRAINING-SET" OLDID="12981"  
NEWID="798">

<DATE> 2-MAR-1987 16:51:43.42</DATE>

<TOPICS><D>livestock</D><D>hog</D></TOPICS>

<TITLE>AMERICAN PORK CONGRESS KICKS OFF TOMORROW</TITLE>

<DATELINE> CHICAGO, March 2 - </DATELINE><BODY>The American Pork Congress kicks off tomorrow, March 3, in Indianapolis with 160 of the nations pork producers from 44 member states determining industry positions on a number of issues, according to the National Pork Producers Council, NPPC.

Delegates to the three day Congress will be considering 26 resolutions concerning various issues, including the future direction of farm policy and the tax law as it applies to the agriculture sector. The delegates will also debate whether to endorse concepts of a national PRV (pseudorabies virus) control and eradication program, the NPPC said.

A large trade show, in conjunction with the congress, will feature the latest in technology in all areas of the industry, the NPPC added. Reuter

<sup>58</sup>  
&#3;</BODY></TEXT></REUTERS>



## Confusion matrix c

- For each pair of classes  $\langle c_1, c_2 \rangle$  how many documents from  $c_1$  were incorrectly assigned to  $c_2$ ?
  - $c_{3,2}$ : 90 wheat documents incorrectly assigned to poultry

Docs in test set	Assigned UK	Assigned poultry	Assigned wheat	Assigned coffee	Assigned interest	Assigned trade
True UK	95	1	13	0	1	0
True poultry	0	1	0	0	0	0
True wheat	10	90	0	1	0	0
True coffee	0	0	0	34	3	7
True interest	-	1	2	13	26	5
True trade	0	0	2	14	5	10



## 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 are actually about class  $i$ :

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

### Accuracy: (1 - error rate)

Fraction of docs classified correctly:

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



## Micro- vs. Macro-Averaging

- If we have more than one class, how do we combine multiple performance measures into one quantity?
- **Macroaveraging:** Compute performance for each class, then average.
- **Microaveraging:** Collect decisions for all classes, compute contingency table, evaluate.



# Micro- vs. Macro-Averaging: Example

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 Ave. Table

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

- Macroaveraged precision:  $(0.5 + 0.9)/2 = 0.7$
- Microaveraged precision:  $100/120 = .83$
- Microaveraged score is dominated by score on common classes



# Development Test Sets and Cross-validation

Training set

Development Test Set

Test Set

- 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
  - Handle sampling errors from different datasets
  - Pool results over each split
  - Compute pooled dev set performance

Training Set Dev Test

Training Set Dev Test

Dev Test Training Set

Test Set







# Text Classification and Naïve Bayes

Text Classification:  
Practical Issues



# The Real World

- Gee, I'm building a text classifier for real, now!
- What should I do?



# No training data? Manually written rules

If (wheat or grain) and not (whole or bread) then  
Categorize as grain

- Need careful crafting
  - Human tuning on development data
  - Time-consuming: 2 days per class



## Very little data?

- Use Naïve Bayes
  - Naïve Bayes is a “high-bias” algorithm (Ng and Jordan 2002 NIPS)
- Get more labeled data
  - Find clever ways to get humans to label data for you
- Try semi-supervised training methods:
  - Bootstrapping, EM over unlabeled documents, ...



## A reasonable amount of data?

- Perfect for all the clever classifiers
  - SVM
  - Regularized Logistic Regression
- You can even use user-interpretable decision trees
  - Users like to hack
  - Management likes quick fixes



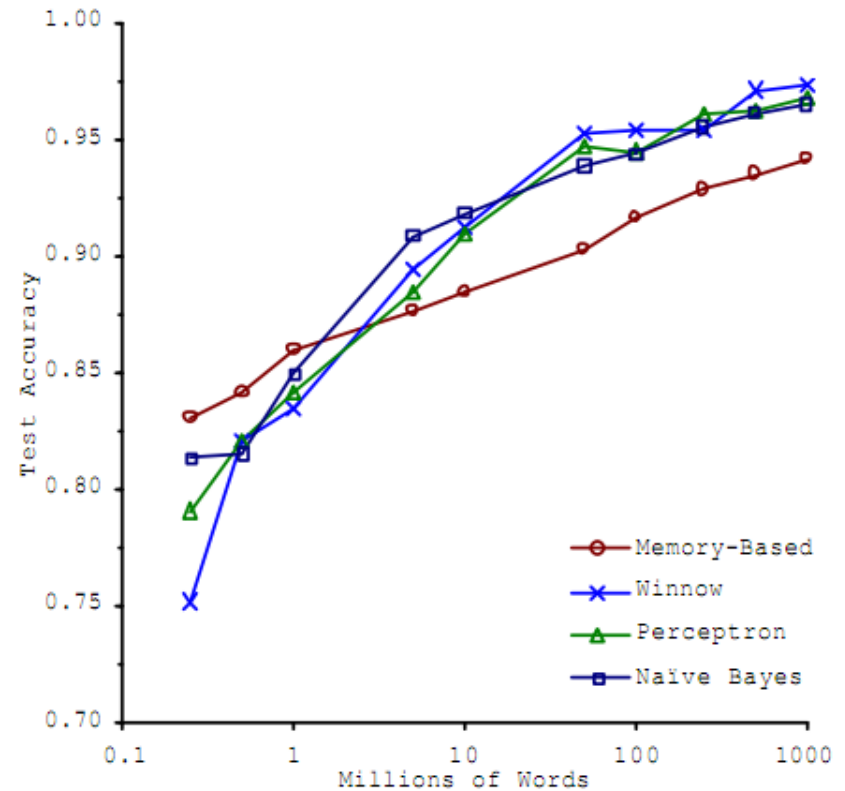
## A huge amount of data?

- Can achieve high accuracy!
- At a cost:
  - SVMs (train time) or kNN (test time) can be too slow
  - Regularized logistic regression can be somewhat better
- So Naïve Bayes can come back into its own again!



# Accuracy as a function of data size

- With enough data
  - Classifier may not matter



Brill and Banko on spelling correction



## Real-world systems generally combine:

- Automatic classification
- Manual review of uncertain/difficult/"new" cases





## Underflow Prevention: log space

- Multiplying lots of probabilities can result in floating-point underflow.
- Since  $\log(xy) = \log(x) + \log(y)$ 
  - Better to sum logs of probabilities instead of multiplying probabilities.
- Class with highest un-normalized log probability score is still most probable.

$$c_{NB} = \operatorname{argmax}_{c_j \in C} \log P(c_j) + \sum_{i \in \text{positions}} \log P(x_i | c_j)$$

- Model is now just max of sum of weights



## How to tweak performance

- Domain-specific features and weights: *very* important in real performance
- Sometimes need to collapse terms:
  - Part numbers, chemical formulas, ...
  - But stemming generally doesn't help
- Upweighting: Counting a word as if it occurred twice:
  - title words (Cohen & Singer 1996)
  - first sentence of each paragraph (Murata, 1999)
  - In sentences that contain title words (Ko *et al*, 2002)

