Text Mining: An Overview

David Madigan

madigan@yahoo.com

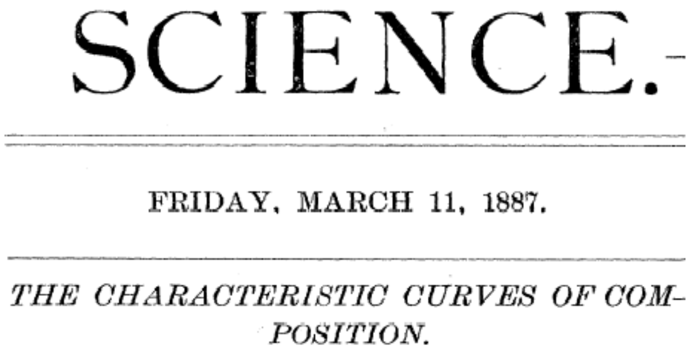
http://www.stat.columbia.edu/~madigan

in collaboration with:

David D. Lewis

Text Mining

- Statistical text analysis has a long history in literary analysis and in solving disputed authorship problems
- First (?) is Thomas C. Mendenhall in 1887



SCIENCE.

FRIDAY, MARCH 11, 1887.

THE CHARACTERISTIC CURVES OF COMPOSITION.

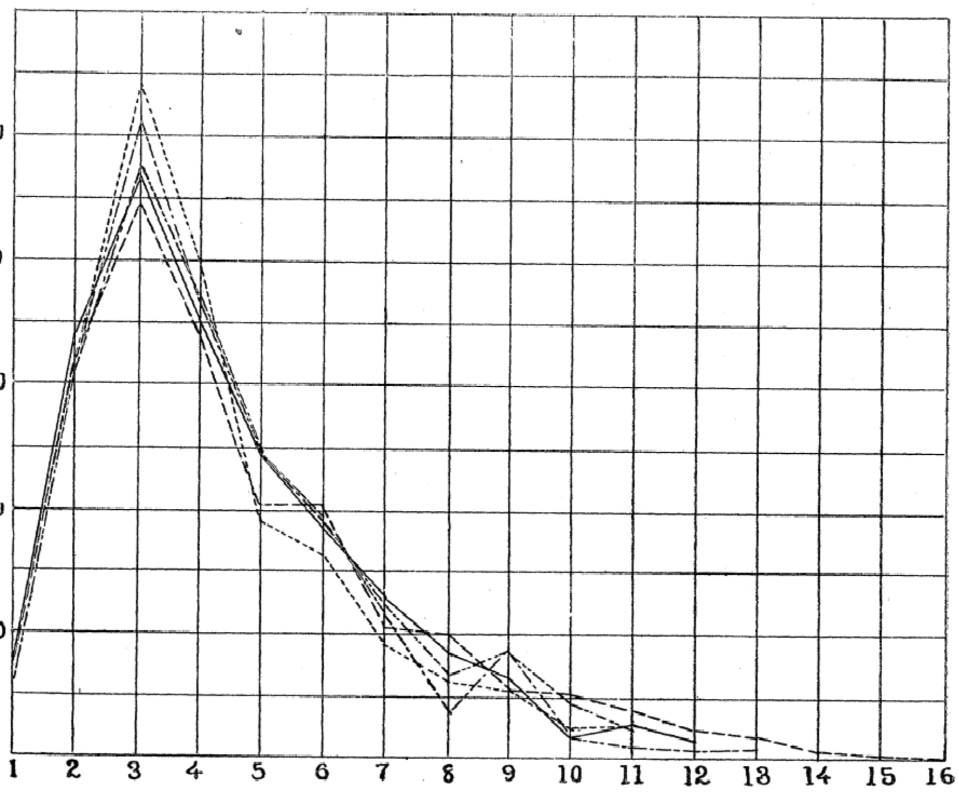


FIG. 2.—SHOWING FIVE GROUPS, OF ONE THOUSAND WORDS EACH, FROM 'OLIVER TWIST.'

Mendenhall

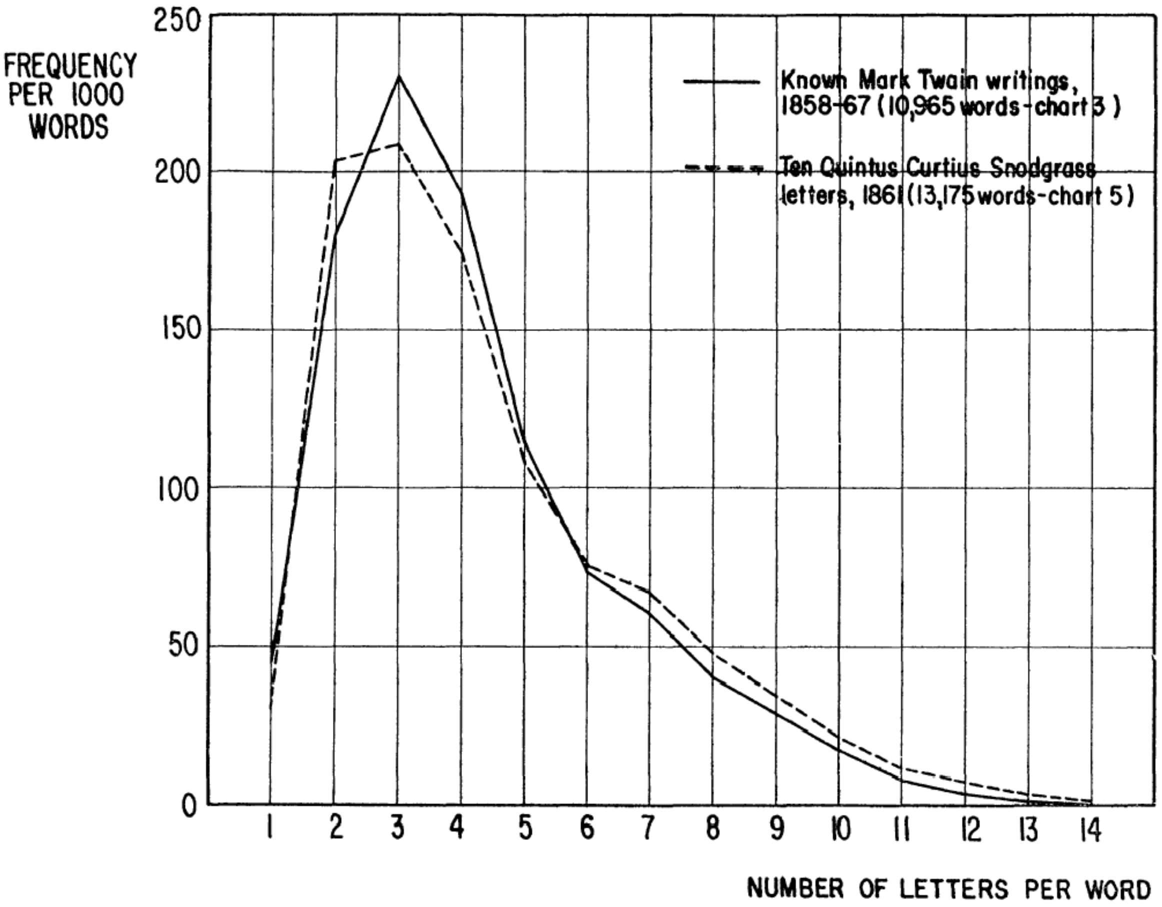
- Mendenhall was Professor of Physics at Ohio State and at University of Tokyo, Superintendent of the USA Coast and Geodetic Survey, and later, President of Worcester Polytechnic Institute

Mendenhall Glacier,
Juneau, Alaska

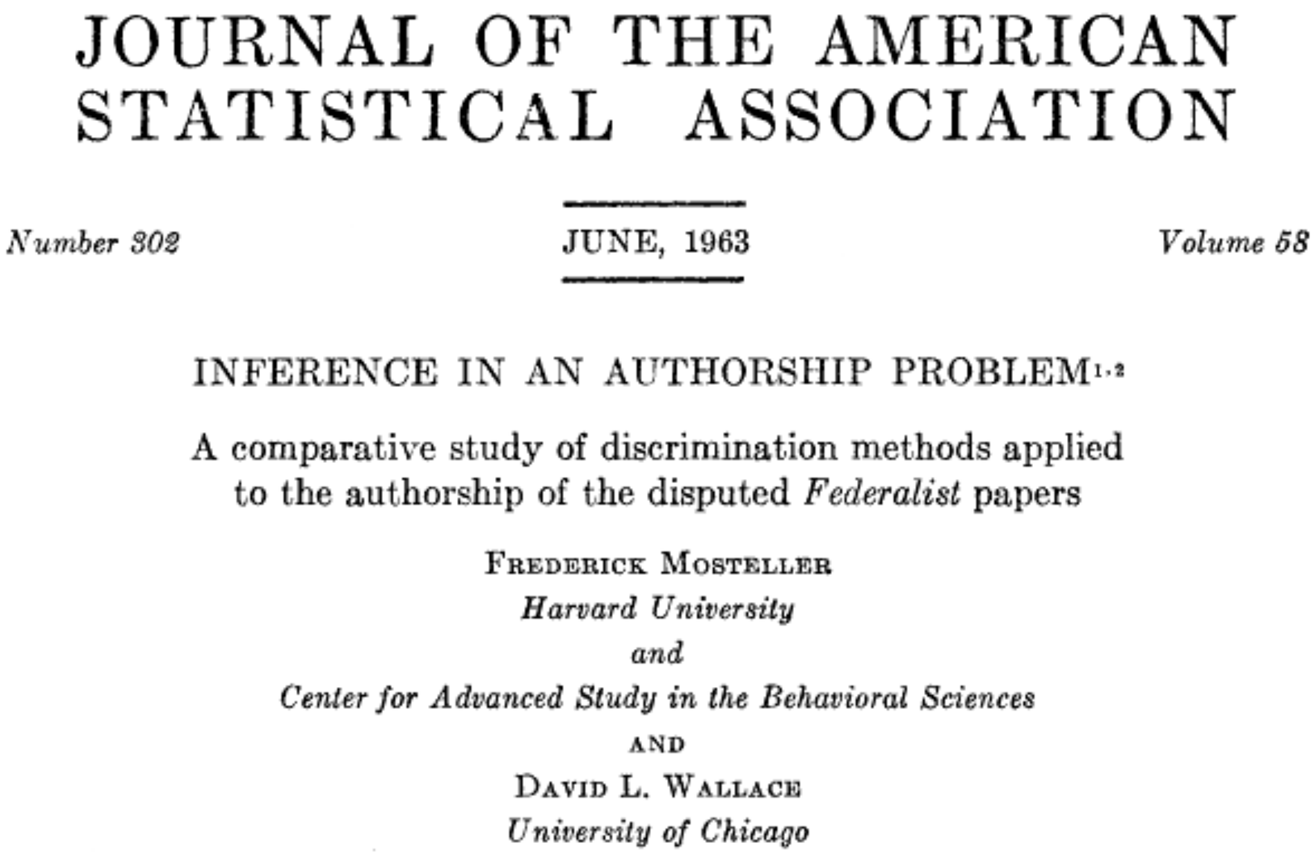
AMERICAN STATISTICAL ASSOCIATION JOURNAL, MARCH 1963

MARK TWAIN AND THE QUINTUS CURTIUS SNODGRASS LETTERS: A STATISTICAL TEST OF AUTHORSHIP

Claude S. Brinegar



X^2 = 127.2, df=12



JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION

Number 302 JUNE, 1963 *Volume 58*

INFERENCE IN AN AUTHORSHIP PROBLEM[1,2]

A comparative study of discrimination methods applied to the authorship of the disputed *Federalist* papers

FREDERICK MOSTELLER
Harvard University
and
Center for Advanced Study in the Behavioral Sciences
AND
DAVID L. WALLACE
University of Chicago

- Hamilton versus Madison
- Used Naïve Bayes with Poisson and Negative Binomial model
- Out-of-sample predictive performance

Today

- Statistical methods routinely used for textual analyses of all kinds
- Machine translation, part-of-speech tagging, information extraction, question-answering, text categorization, disputed authorship (stylometry), etc.
- Not reported in the statistical literature (no statisticians?)

Mosteller, Wallace, Efron, Thisted

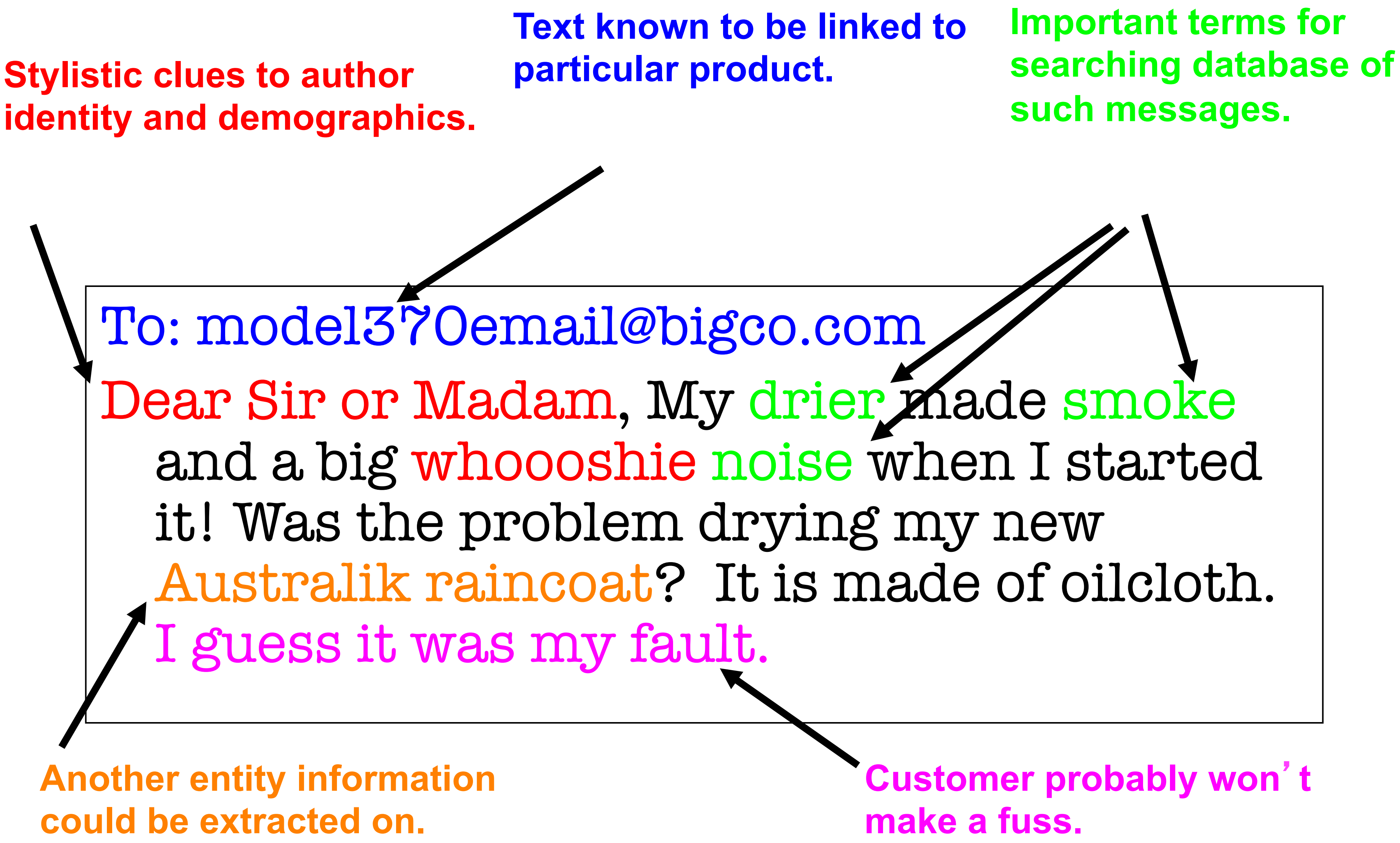
Text Mining' s Connections with Language Processing

- Linguistics
- Computational linguistics
- Information retrieval
- Content analysis
- Stylistics
- Others

Why Are We Mining the Text?

Are we trying to understand:

1. The texts themselves?
2. The writer (or speakers) of the texts?
    a. The writer as a writer?
    b. The writer as an entity in the world?
3. Things in the world?
    a. Directly linked to texts?
    b. Described by texts?



Stylistic clues to author identity and demographics.

Text known to be linked to particular product.

Important terms for searching database of such messages.

To: model370email@bigco.com

Dear Sir or Madam, My drier made smoke and a big whoooshie noise when I started it! Was the problem drying my new Australik raincoat? It is made of oilcloth. I guess it was my fault.

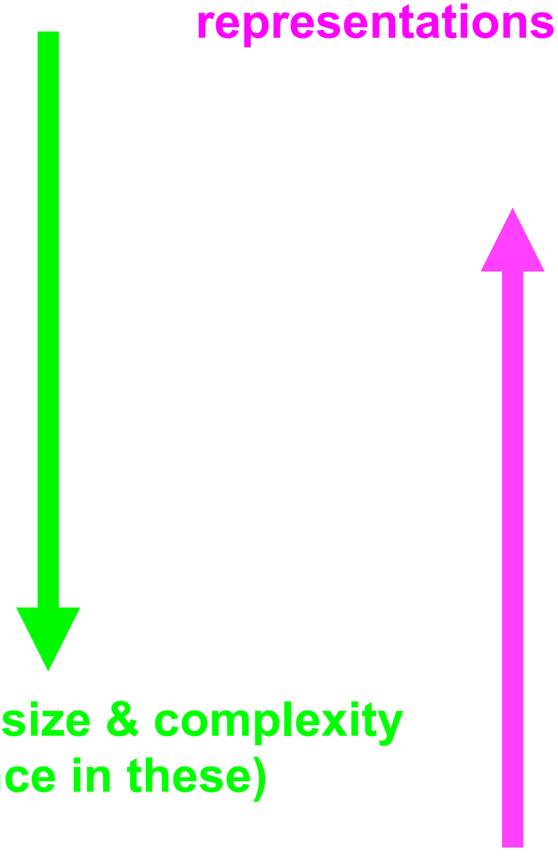
Another entity information could be extracted on.

Customer probably won’t make a fuss.

Granularity of Text Linked to Entity?

- Morphemes, words, simple noun phrases
- Clauses, sentences
- Paragraphs, sections, documents
- Corpora, networks of documents

Increasing linguistic agreement on structure and representations



Increasing size & complexity (and variance in these)

Ambiguity

- Correct interpretation of an utterance is rarely explicit!
  - People use massive knowledge of language and the world to understand NL
  - information readily inferred by speaker/reader will be left out
- ***The core task in NLP is resolving the resulting ambiguity***

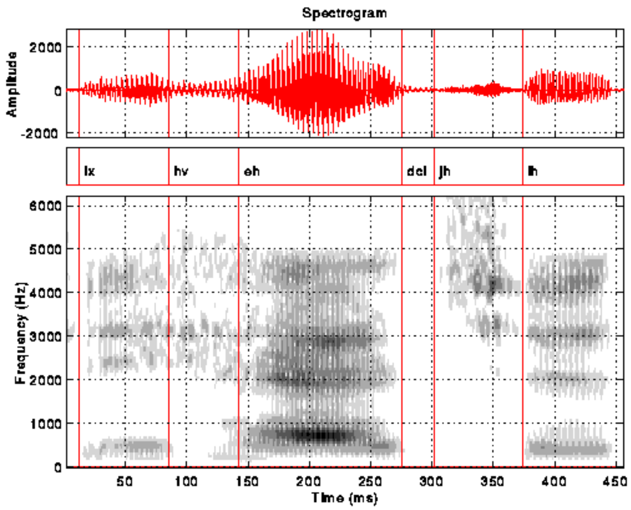
"I made her duck" [after Jurafsky & Martin]

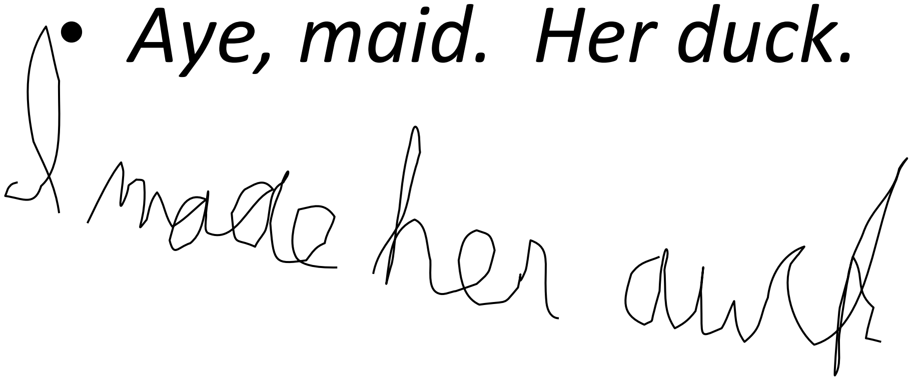
- *I cooked waterfowl for her.*
- *I cooked waterfowl belonging to her.*
- *I created the (plaster?) duck she owns.*
- *I caused her to quickly lower her head.*
- *I magically converted her into roast fowl.*

- These vary in morphology, syntax, semantic, pragmatics.

....Plus, Phonetic and Visual Ambiguity

- *Aye, made her duck!*
- *<u>I</u> made her....duck!*
- *Aye, maid. Her duck.*





Synonymy

- Can use different words to communicate the same meaning
  - (Of course also true for sentences,....)
- Synonymy in all contexts is rare:
  - *a big plane = a large plane*
  - *a big sister ≠ a large sister*
- And choice among synonyms may be a clue to topic, writer' s background, etc.
  - *rich* vs. *high SES*

Metaphor & Metonymy

- Metaphor: use of a construct with one meaning to convey a very different one:
  - *Television is eating away at the moral fiber of our country.*
- Metonymy: mentioning one concept to convey a closely related one

  - *"On the way downtown I stopped at a bar and had a couple of double Scotches. They didn't do me any good. All they did was make me think of Silver Wig, and I never saw her again."*

  (Raymond Chandler, The Big Sleep)

Attribute Vectors

- Most text mining based on
  - Breaking language into symbols
  - Treating each symbol as an attribute
- But what value should each attribute have for a unit of text?

Term Weighting

- How strongly does a particular word indiciate the content of a document?
- Some clues:
  - Number of times word occurs in this document
  - Number of times word occurs in other documents
  - Length of document

TF (term frequency)

IDF (inverse document frequency)

$$w_{ij}^{\text{raw}} = \begin{matrix} (1 + \ln f_{ij}) \ln \dfrac{N}{n_j}, & \text{if } t_j \text{ present in } d_i \\ 0, & \text{otherwise} \end{matrix}$$

$$w_{ij} = \frac{w_{ij}^{\text{raw}}}{\sqrt{\sum_{j'=1}^{d} w_{ij}^{\text{raw}} \times w_{ij}^{\text{raw}}}}$$

Set L2-norm to 1.0

- "Cosine-normalized TFIDF weighting"
  - Many minor variants on this theme

Case Study: Representation for Authorship Attribution

- Statistical methods for authorship attribution
- Represent documents with attribute vectors
- Then use regression-type methods
- Bag of words?
- Stylistic features? (e.g., passive voice)
- Topic free?

1-of-K Sample Results: brittany-l

Feature Set	% errors	Number of Features
“Argamon” function words, raw tf	74.8	380
POS	75.1	44
1suff	64.2	121
1suff*POS	50.9	554
2suff	40.6	1849
2suff*POS	34.9	3655
3suff	28.7	8676
3suff*POS	27.9	12976
3suff+POS+3suff*POS +Argamon	27.6	22057
All words	23.9	52492

4.6 million parameters

89 authors with at least 50 postings. 10,076 training documents, 3,322 test documents.

BMR-Laplace classification, default hyperparameter

Features	Name in Short
The length of each word	charcount
Part of speeches	POS
Two-letter-suffix	Suffix2
Three-letter-suffix	Suffix3
Words, numbers, signs, punctuations	Words
The length of each word plus part of speech tags	Charcount+POS
Two-letter-suffix plus part of speech tags	Suffix2+POS
Three-letter-suffix plus part of speech tags	Suffix3+POS
Words, numbers, signs, punctuations plus part of speech tags	Words+POS
484 function words from Koppel et al's paper	484 features
Mosteller and Wallace function words	Wallace features
Words appear at least twice	Words(¿=2)
Every word in the Federalist papers	Each word

The Federalist

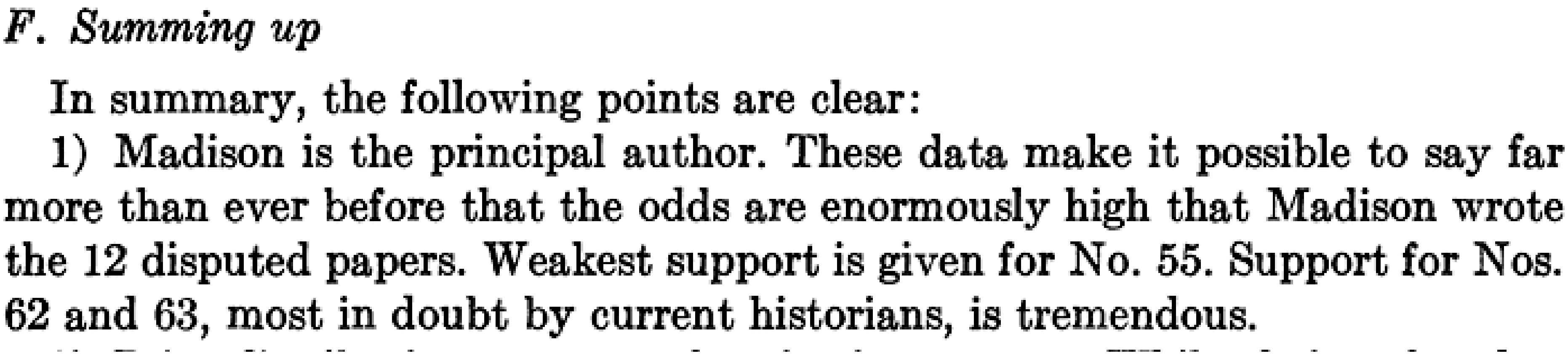
- Mosteller and Wallace attributed all 12 disputed papers to Madison
- Historical evidence is more muddled
- Our results suggest attribution is highly dependent on the document representation

Table 1 Authorship of the Federalist Papers

Paper Number	Author
1	Hamilton
2-5	Jay
6-9	Hamilton
10	Madison
11-13	Hamilton
14	Madison
15-17	Hamilton
18-20	Joint: Hamilton and Madison
21-36	Hamilton
37-48	Madison
49-58	**Disputed**
59-61	Hamilton
62-63	**Disputed**
64	Jay
65-85	Hamilton

Table 3 The feature sets

Features	**Name in Short**
The length of each character	charcount
Part of speeches	POS
Two-letter-suffix	Suffix2
Three-letter-suffix	Suffix3
Words, numbers, signs, punctuations	Words
The length of each character plus the part of speeches	Charcount+POS
Two-letter-suffix plus the part of speeches	Suffix2+POS
Three-letter-suffix plus the part of speeches	Suffix3+POS
Words, numbers, signs, punctuations plus the part of speeches	Words+POS
The 484 function words in Koppel's paper	484 features
The feature set in the Mosteller and Wallace paper	Wallace features
Words appear at least twice	Words(>=2)
Each word shown in the Federalist papers	Each word



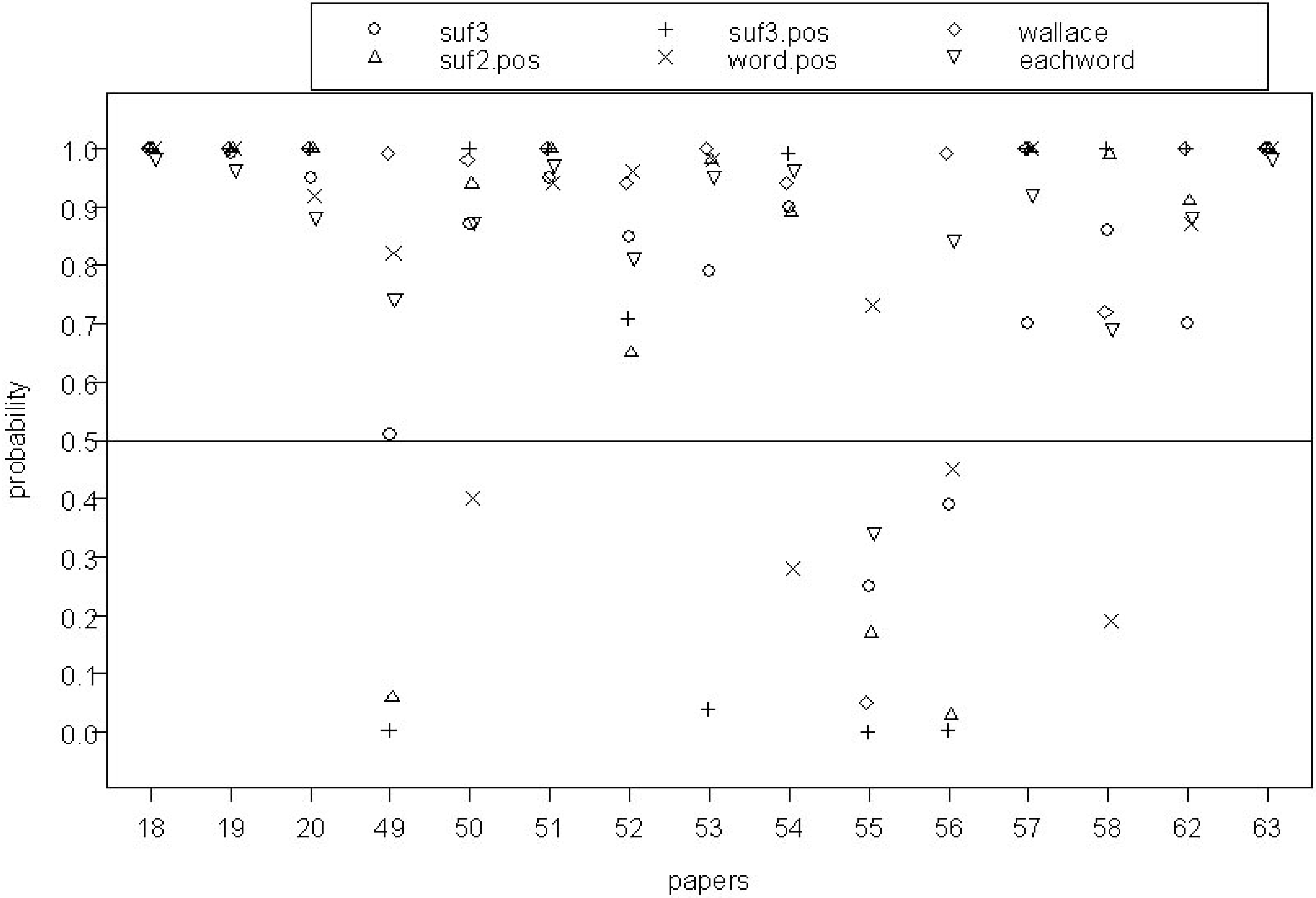
F. Summing up

In summary, the following points are clear:

1) Madison is the principal author. These data make it possible to say far more than ever before that the odds are enormously high that Madison wrote the 12 disputed papers. Weakest support is given for No. 55. Support for Nos. 62 and 63, most in doubt by current historians, is tremendous.

Feature Set	**10-fold Error Rate**
Charcount	0.21
POS	0.19
Suffix2	0.12
Suffix3	0.09
Words	0.10
Charcount+POS	0.12
Suffix2+POS	0.08
Suffix3+POS	0.04
Words+POS	0.08
484 features	0.05
Wallace features	0.05
Words (>=2)	0.05
Each Word	0.05

four papers to Hamilton


○ suf3
+ suf3.pos
◇ wallace
△ suf2.pos
× word.pos
▽ eachword
probability
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
18
19
20
49
50
51
52
53
54
55
56
57
58
62
63
papers

Supervised Learning for Text Classification

Predictive Modeling

Goal: learn a mapping: $y = f(\boldsymbol{x};□)$

Need: 1. A model structure

2. A score function

3. An optimization strategy

Categorical y □ $\{c_1,\ldots,c_m\}$: classification

Real-valued y: regression

Note: usually assume $\{c_1,\ldots,c_m\}$ are mutually exclusive and exhaustive

Classifier Types

Discriminative: model $p(c_k \mid \boldsymbol{x})$

- e.g. linear regression, logistic regression, CART

Generative: model $p(\boldsymbol{x} \mid c_k, \square_k)$

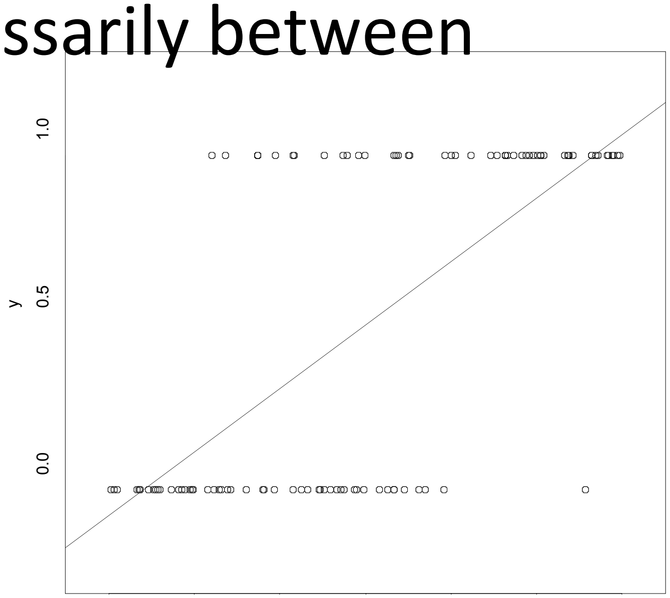
- e.g. “Bayesian classifiers”, LDA

Regression for Binary Classification

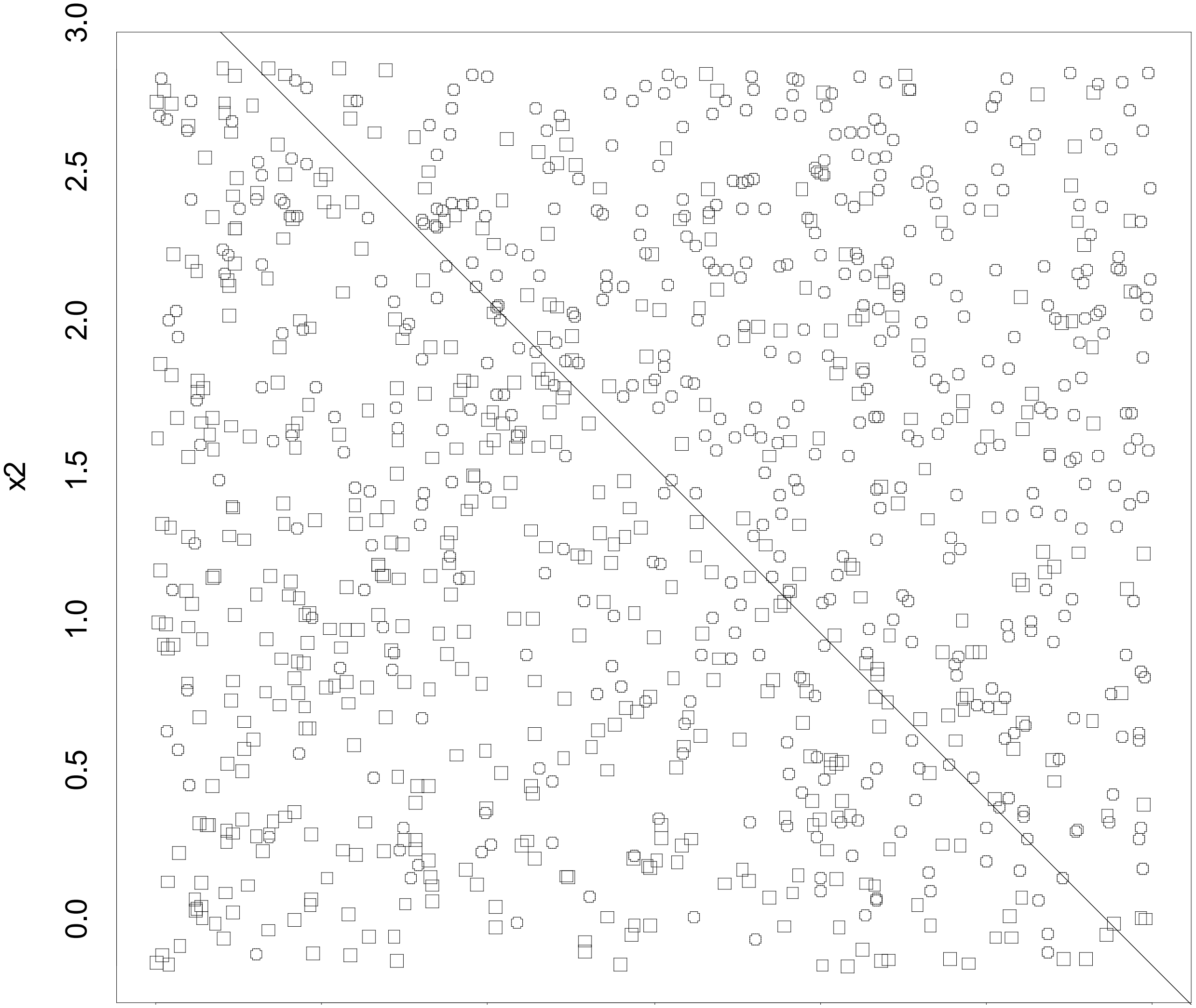
- Can fit a linear regression model to a 0/1 response

- Predicted values are not necessarily between zero and one

- With p>1, the decision boundary is linear
  e.g. 0.5 = b0 + b1 x1 + b2 x2



zeroOneR.txt


x2
0.0
0.5
1.0
1.5
2.0
2.5
3.0

Naïve Bayes via a Toy Spam Filter Example

- Naïve Bayes is a generative model that makes drastic simplifying assumptions
- Consider a small training data set for spam along with a bag of words representation

#	**Message**	**Spam**
1	the quick brown fox	no
2	the quick rabbit ran and ran	yes
3	rabbit run run run	no
4	rabbit at rest	yes

Training data comprising four labeled e-mail messages.

#	**and**	**at**	**brown**	**fox**	**quick**	**rabbit**	**ran**	**rest**	**run**	**the**
1	0	0	1	1	1	0	0	0	0	1
2	1	0	0	0	1	1	2	0	0	1
3	0	0	0	0	0	1	0	0	3	0
4	0	1	0	0	0	1	0	1	0	0

Term vectors corresponding to the training data.

	X_1	X_2	X_3	X_4	X_5	X_6	Y
#	**brown**	**fox**	**quick**	**rabbit**	**rest**	**run**	**Spam**
1	1	1	1	0	0	0	0
2	0	0	1	1	0	2	1
3	0	0	0	1	0	3	0
4	0	0	1	1	1	0	1

Term vectors after stemming and stopword removal with the Spam label, coded as 0=no, 1=yes.

Naïve Bayes Machinery

- We need a way to estimate:

$$Pr(Y=1|X_1=x_1,\ldots,X_d=x_d)$$

- Via Bayes theorem we have:

$$=\frac{Pr(Y=1)\times Pr(X_1=x_1,\ldots,X_d=x_d|Y=1)}{Pr(X_1=x_1,\ldots,X_d=x_d)}$$

or, on the log-odds scale:

$$\log\frac{Pr(Y=1|X_1=x_1,\ldots,X_d=x_d)}{Pr(Y=0|X_1=x_1,\ldots,X_d=x_d)}$$

$$=\log\frac{Pr(Y=1)}{Pr(Y=0)}+\log\frac{Pr(X_1=x_1,\ldots,X_d=x_d|Y=1)}{Pr(X_1=x_1,\ldots,X_d=x_d|Y=0)}$$

Naïve Bayes Machinery

- Naïve Bayes assumes:

$$Pr(X_1 = x_1, \ldots, X_d = x_d | Y = 1) = \prod_{i=1}^{d} Pr(X_i = x_i | Y = 1)$$

and

$$Pr(X_1 = x_1, \ldots, X_d = x_d | Y = 0) = \prod_{i=1}^{d} Pr(X_i = x_i | Y = 0)$$

leading to:

$$\log \frac{Pr(Y = 1 | X_1 = x_1, \ldots, X_d = x_d)}{Pr(Y = 0 | X_1 = x_1, \ldots, X_d = x_d)}$$

$$= \log \frac{Pr(Y = 1)}{Pr(Y = 0)} + \sum_{i=1}^{d} \log \frac{Pr(X_i = x_i | Y = 1)}{Pr(X_i = x_i | Y = 0)}$$

Maximum Likelihood Estimation

$$\log \frac{\widehat{Pr}(Y=1)}{\widehat{Pr}(Y=0)} = \log \frac{2/4}{2/4} = 0$$

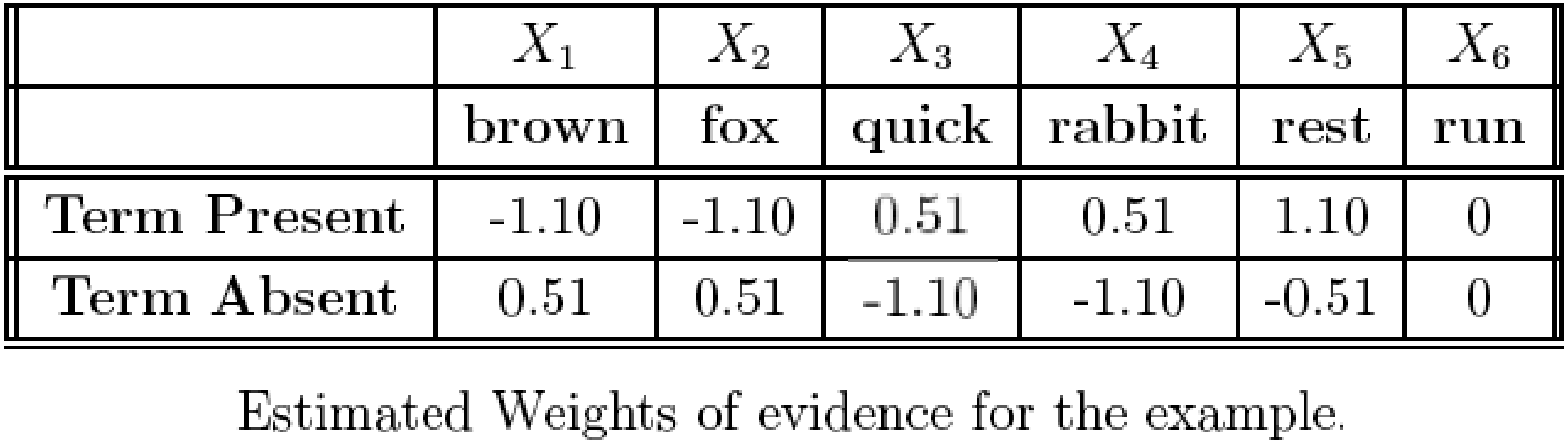
weights of evidence

$$\log \frac{\widehat{Pr}(X_3=1|Y=1)}{\widehat{Pr}(X_3=1|Y=0)} = \log \frac{2/2}{1/2} = \log 2$$

	X_1	X_2	X_3	X_4	X_5	X_6	Y
#	brown	fox	quick	rabbit	rest	run	Spam
1	1	1	1	0	0	0	0
2	0	0	1	1	0	2	1
3	0	0	0	1	0	3	0
4	0	0	1	1	1	0	1

Naïve Bayes Prediction

- Usually add a small constant (e.g. 0.5) to avoid divide by zero problems and to reduce bias

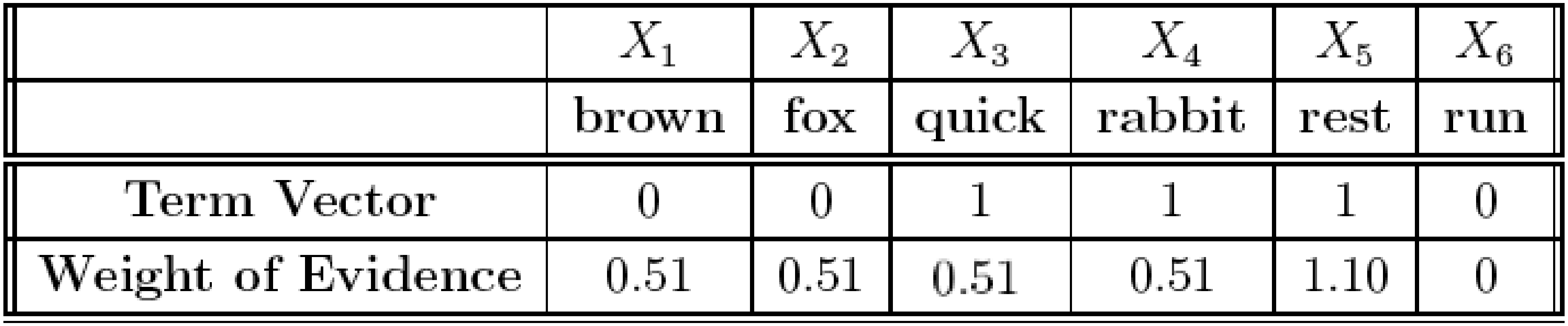


	X_1	X_2	X_3	X_4	X_5	X_6
	brown	fox	quick	rabbit	rest	run
Term Present	-1.10	-1.10	0.51	0.51	1.10	0
Term Absent	0.51	0.51	-1.10	-1.10	-0.51	0

Estimated Weights of evidence for the example.

- New message: “the quick rabbit rests”

- New message: “the quick rabbit rests”



	X_1	X_2	X_3	X_4	X_5	X_6
	brown	fox	quick	rabbit	rest	run
Term Vector	0	0	1	1	1	0
Weight of Evidence	0.51	0.51	0.51	0.51	1.10	0

- Predicted log odds:

$$0.51 + 0.51 + 0.51 + 0.51 + 1.10 + 0 = 3.04$$

- Corresponds to a spam probability of 0.95

A Close Look at Logistic Regression for Text Classification

Logistic Regression

- Linear model for log odds of category membership:

$$\log \frac{p(\mathrm{y}=1 \mid \boldsymbol{x}_i)}{p(\mathrm{y}=-1 \mid \boldsymbol{x}_i)} = \sum b_j\, x_{ij} = \boldsymbol{bx}_i$$

Maximum Likelihood Training

- Choose parameters (*b*$_j$'s) that maximize probability (likelihood) of class labels (*y*$_i$'s) given documents (***x***$_i$'s)

$$L(\boldsymbol{\beta}) = p(\boldsymbol{\beta}|D) = (\prod_{i=1}^{n} \frac{1}{1 + \exp(-\boldsymbol{\beta}^T \boldsymbol{x_i} y_i)})$$

- Tends to overfit
- Not defined if *d* > *n*
- Feature selection

Shrinkage/Regularization/Bayes

- Avoid combinatorial challenge of feature selection
- L1 shrinkage: regularization + feature selection
- Expanding theoretical understanding
- Large scale
- Empirical performance

Ridge Logistic Regression

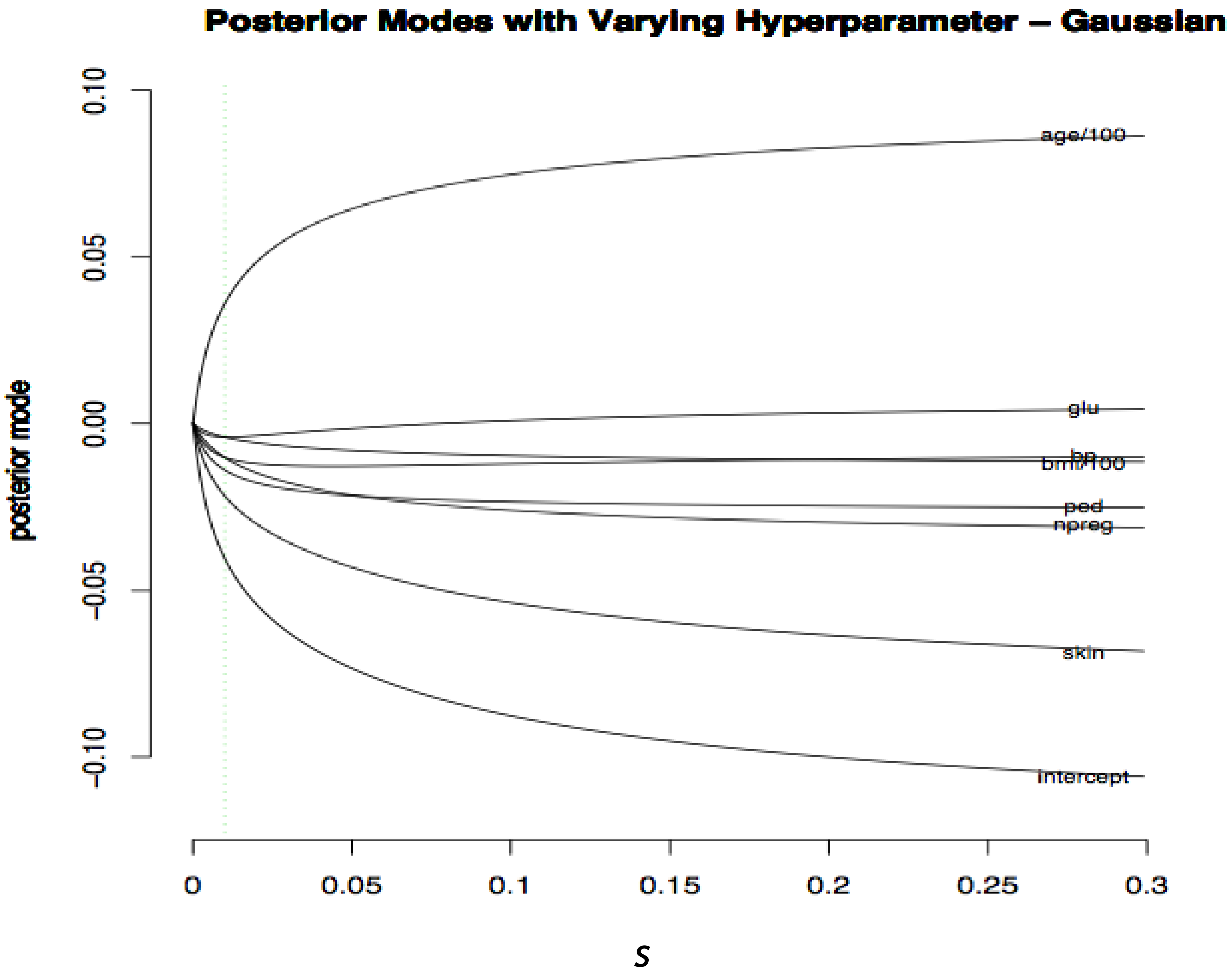
Maximum likelihood plus a constraint:

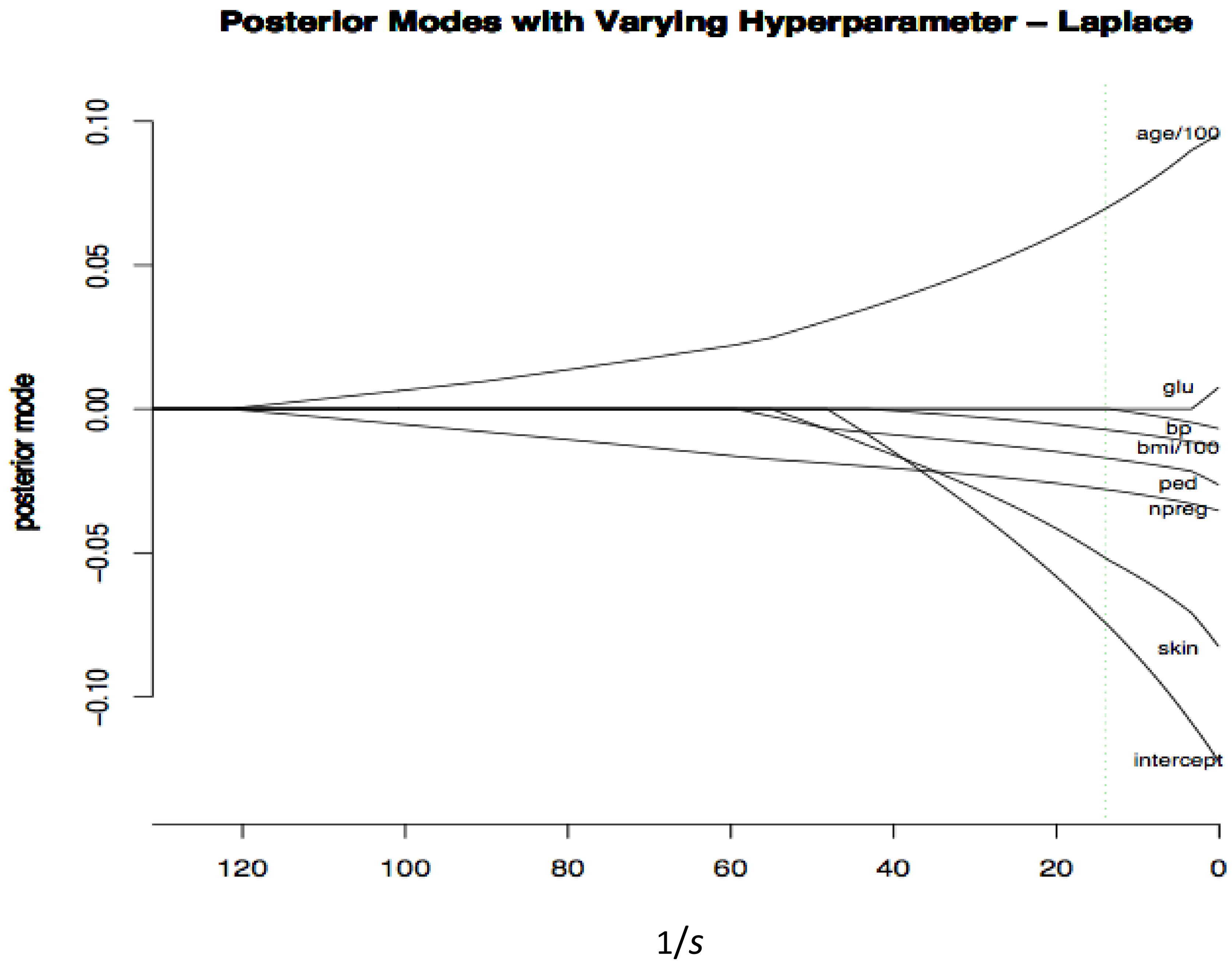
$$\sum_{j=1}^{p} \beta_j^2 \le s$$

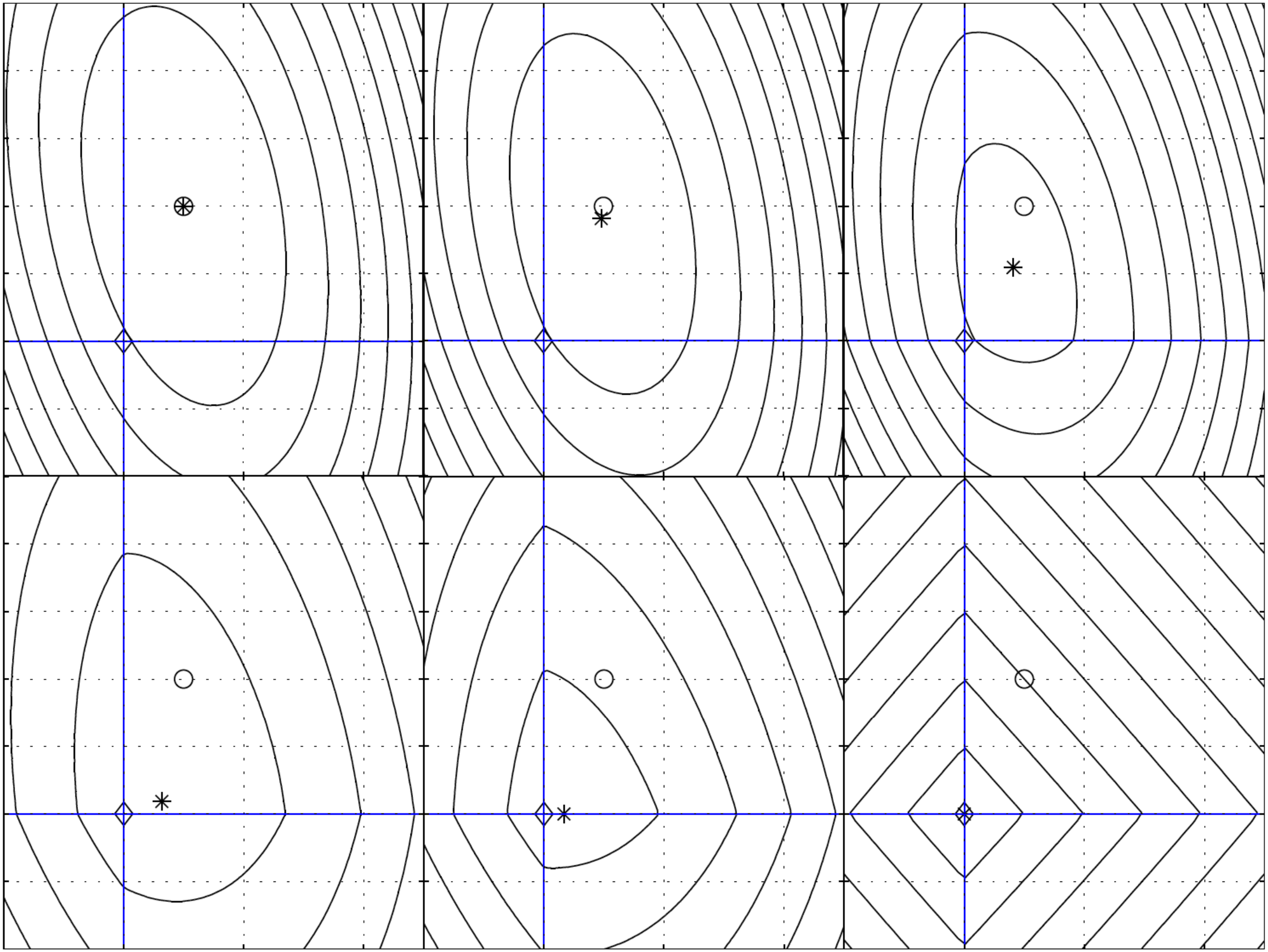
Lasso Logistic Regression

Maximum likelihood plus a constraint:

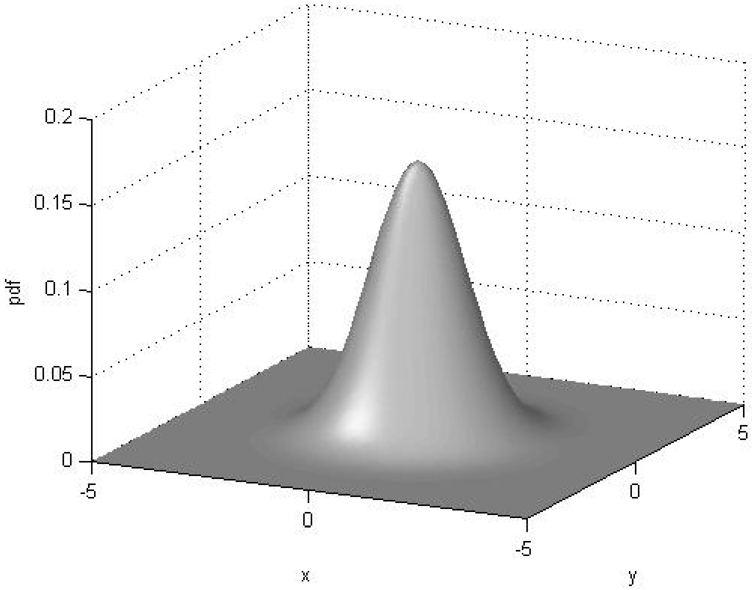
$$\sum_{j=1}^{p} \left|\beta_j\right| \le s$$


Posterior Modes with Varying Hyperparameter – Gaussian
posterior mode
0.10
0.05
0.00
−0.05
−0.10
age/100
glu
bp
bmi/100
ped
npreg
skin
intercept
0
0.05
0.1
0.15
0.2
0.25
0.3
S

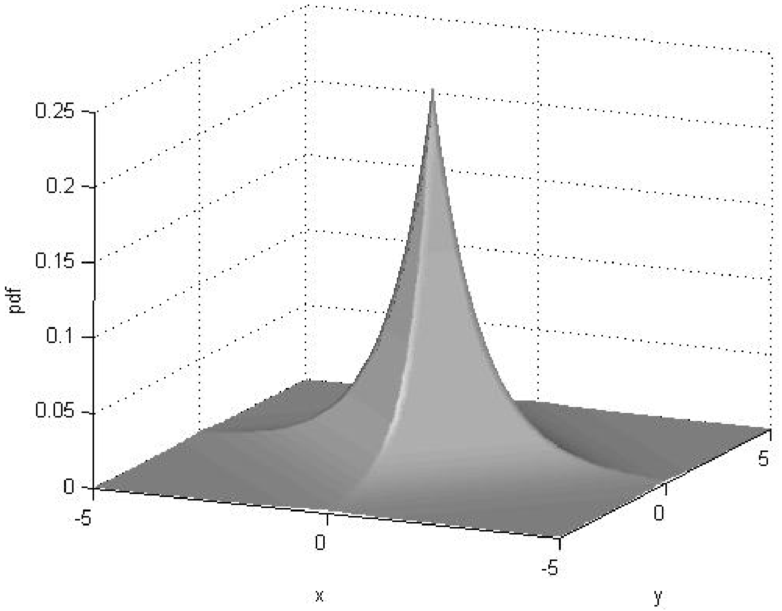

Posterior Modes with Varying Hyperparameter – Laplace
posterior mode
0.10
0.05
0.00
-0.05
-0.10
120
100
80
60
40
20
0
1/*s*
age/100
glu
bp
bmi/100
ped
npreg
skin
intercept



Bayesian Perspective



$$\beta_j \sim N(0, \tau^2)$$



$$\beta_j \sim N(0, \tau_j^2)$$

$$\tau_j^2 \sim \exp(\gamma)$$

Polytomous Logistic Regression (PLR)

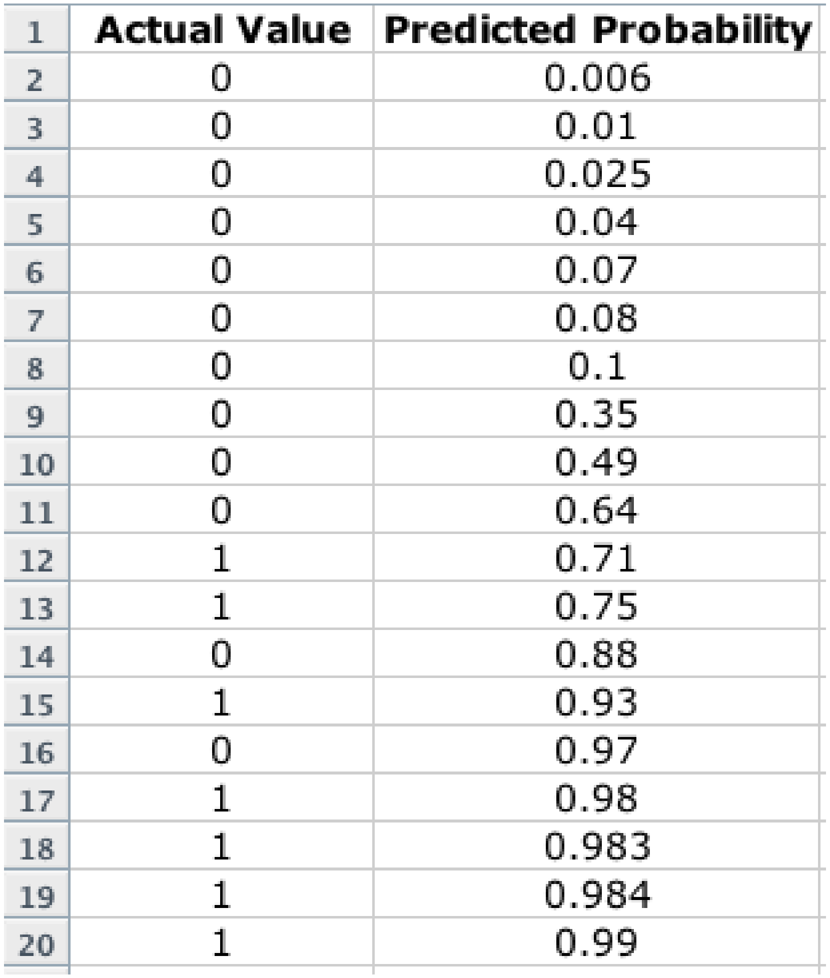
$$P(y_i = k \mid \mathbf{x_i}) = \frac{\exp(\vec{\beta}_k \mathbf{x}_i)}{\sum_{k'} \exp(\vec{\beta}_{k'} \mathbf{x}_i)}$$

- Elegant approach to multiclass problems
- Also known as *polychotomous LR*, *multinomial LR*, and, ambiguously, *multiple LR* and *multivariate LR*

Why LR is Interesting

- Parameters have a meaning
  - How log odds increases w/ feature values
- Lets you
  - Look at model and see if sensible
  - Use domain knowledge to guide parameter fitting (more later)
  - Build some parts of model by hand
- Cavaet: realistically, a lot can (does) complicate this interpretation

Measuring the Performance of a Binary Classifier



1	Actual Value	Predicted Probability
2	0	0.006
3	0	0.01
4	0	0.025
5	0	0.04
6	0	0.07
7	0	0.08
8	0	0.1
9	0	0.35
10	0	0.49
11	0	0.64
12	1	0.71
13	1	0.75
14	0	0.88
15	1	0.93
16	0	0.97
17	1	0.98
18	1	0.983
19	1	0.984
20	1	0.99

Test Data

Suppose we use a cutoff of 0.5…

		actual outcome	
		1	0
predicted outcome	1	7	3
	0	0	10

More generally…

		actual outcome	
		1	0
predicted outcome	1	a	b
	0	c	d

misclassification rate: $\dfrac{b + c}{a+b+c+d}$

sensitivity: $\dfrac{a}{a+c}$

(aka recall)

specificity: $\dfrac{d}{b+d}$

predicitive value positive: $\dfrac{a}{a+b}$

(aka precision)

Suppose we use a cutoff of 0.5…

		actual outcome	
		1	0
predicted outcome	1	7	3
	0	0	10

sensitivity: $\frac{7}{7+0}$ = 100%

specificity: $\frac{10}{10+3}$ = 77%

Suppose we use a cutoff of 0.8…

		actual outcome	
		1	0
predicted outcome	1	5	2
	0	2	11

sensitivity: $\frac{5}{5+2}$ = 71%

specificity: $\frac{11}{11+2}$ = 85%

- Note there are 20 possible thresholds

- ROC computes sensitivity and specificity for all possible thresholds and plots them

- Note if threshold = minimum

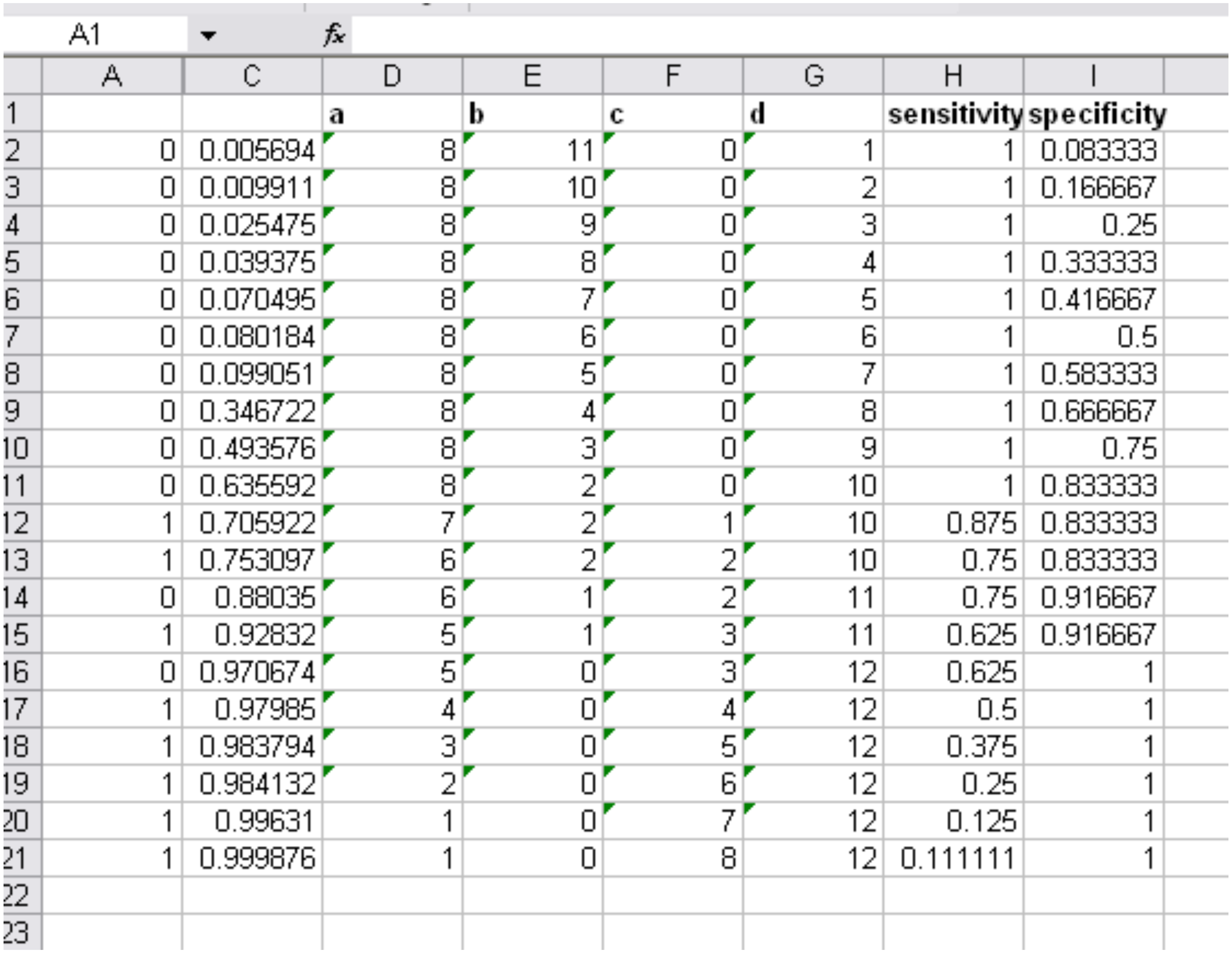
  c=*d*=0 so sens=1; spec=0

- If threshold = maximum

  a=*b*=0 so sens=0; spec=1

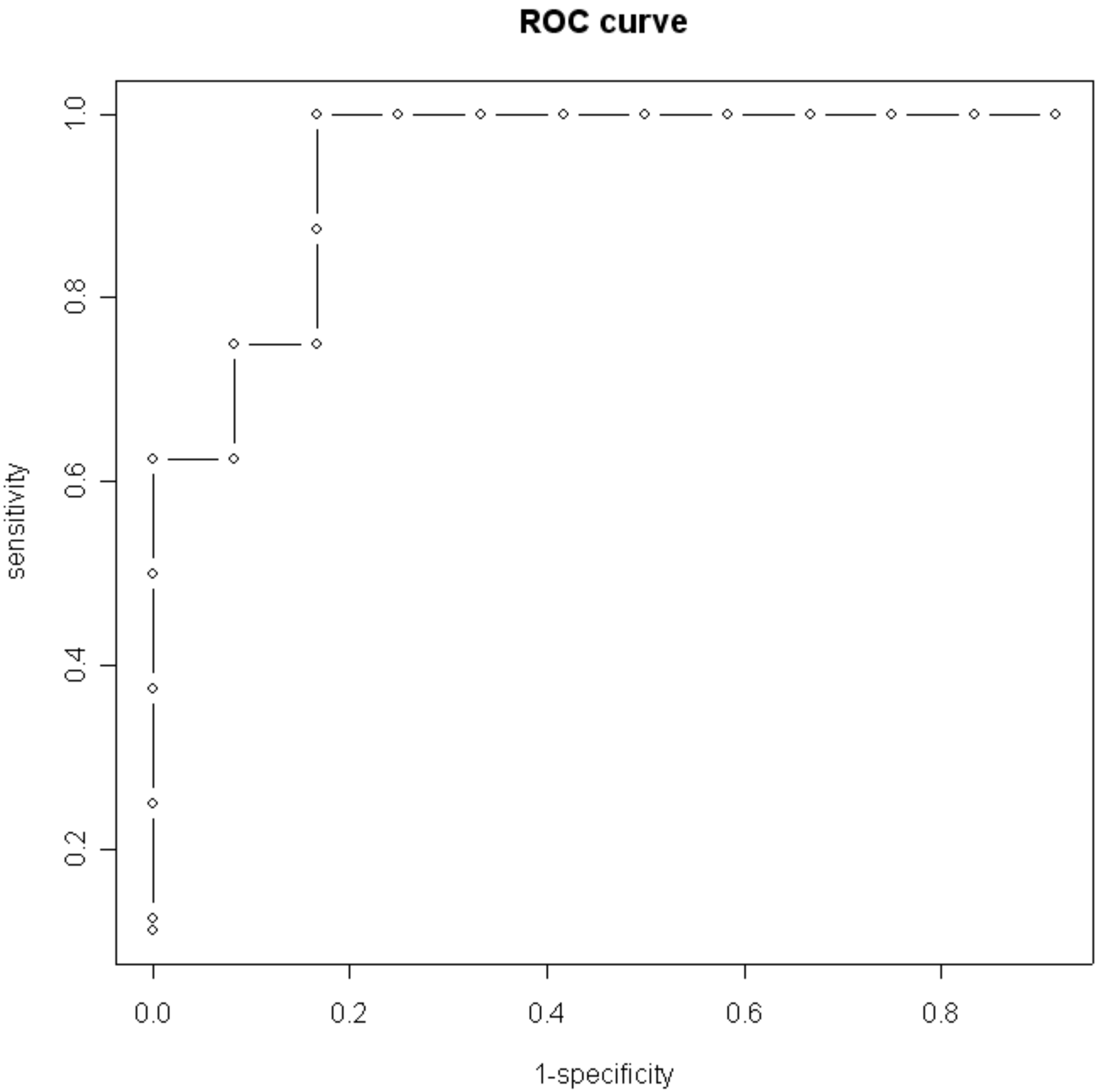
actual outcome

	1	0
1	*a*	*b*
0	*c*	*d*



	A	C	D	E	F	G	H	I
1			a	b	c	d	sensitivity	specificity
2	0	0.005694	8	11	0	1	1	0.083333
3	0	0.009911	8	10	0	2	1	0.166667
4	0	0.025475	8	9	0	3	1	0.25
5	0	0.039375	8	8	0	4	1	0.333333
6	0	0.070495	8	7	0	5	1	0.416667
7	0	0.080184	8	6	0	6	1	0.5
8	0	0.099051	8	5	0	7	1	0.583333
9	0	0.346722	8	4	0	8	1	0.666667
10	0	0.493576	8	3	0	9	1	0.75
11	0	0.635592	8	2	0	10	1	0.833333
12	1	0.705922	7	2	1	10	0.875	0.833333
13	1	0.753097	6	2	2	10	0.75	0.833333
14	0	0.88035	6	1	2	11	0.75	0.916667
15	1	0.92832	5	1	3	11	0.625	0.916667
16	0	0.970674	5	0	3	12	0.625	1
17	1	0.97985	4	0	4	12	0.5	1
18	1	0.983794	3	0	5	12	0.375	1
19	1	0.984132	2	0	6	12	0.25	1
20	1	0.99631	1	0	7	12	0.125	1
21	1	0.999876	1	0	8	12	0.111111	1
22								
23								

```
sens<-c(1,1,1,1,1,1,1,1,1,1,0.875,0.75,0.75,0.625,0.625,0.5,0.375,0.25,0.125,0.11111
spec<-c(0.083333333,0.166666667,0.25,0.333333333,0.416666667,0.5,0.583333333,0.66666
33333,0.916666667,0.916666667,1,1,1,1,1,1)
plot(1-spec,sens,type="b",xlab="1-specificity",ylab="sensitivity",main="ROC curve")
```


ROC curve
sensitivity
1-specificity
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8

- “Area under the curve” is a common measure of predictive performance

- So is squared error: $S(y_i\text{-}y\text{hat})^2$

  also known as the “Brier Score”

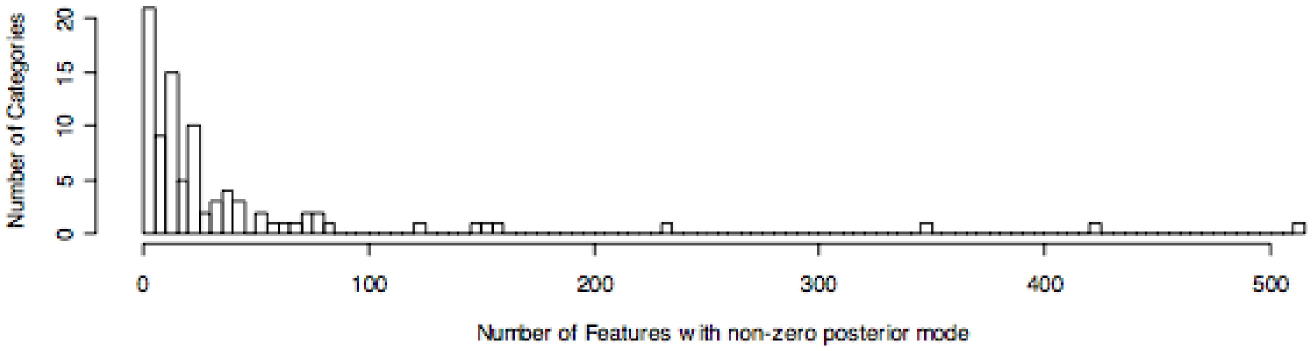
Text Classification Example

- ModApte subset of Reuters-21578
  - 90 categories; 9603 training docs; 18978 features
- Reuters RCV1-v2
  - 103 cats; 23149 training docs; 47152 features
- OHSUMED heart disease categories
  - 77 cats; 83944 training docs; 122076 features
- Cosine normalized TFxIDF weights

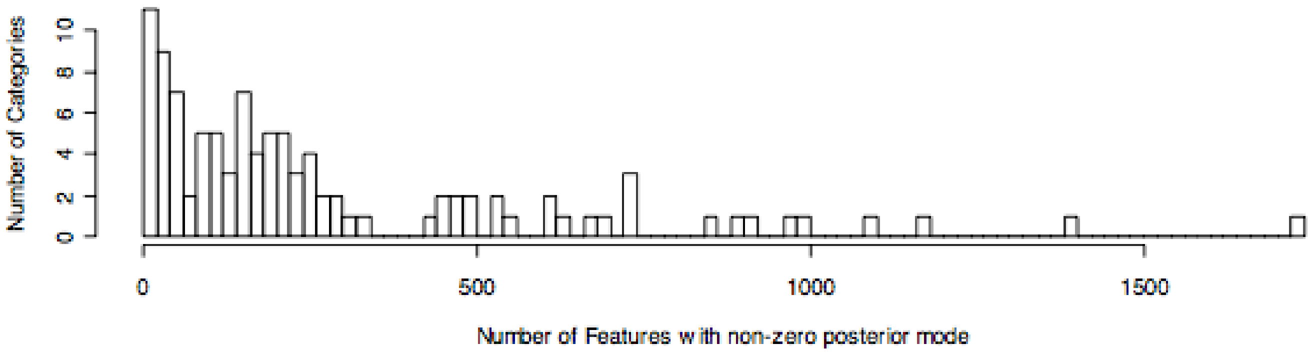
Dense vs. Sparse Models (Macroaveraged F1)

	ModApte	RCV1-v2	OHSUMED
Lasso	**52.03**	**56.54**	**51.30**
Ridge	39.71	51.40	42.99
Ridge/500	38.82	46.27	36.93
Ridge/50	45.80	41.61	42.59
Ridge/5	46.20	28.54	41.33
SVM	53.75	57.23	50.58

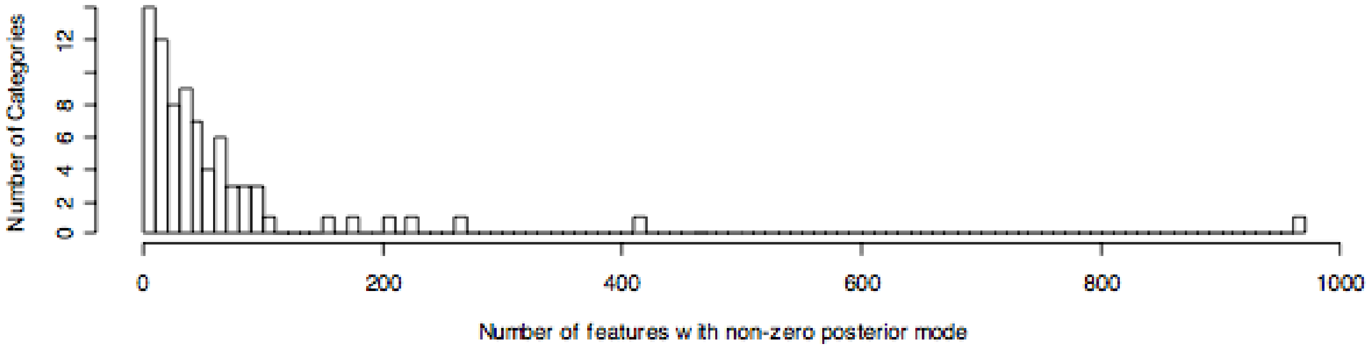
ModApte - 21,989 features



RCV1 - 47,152 features



OHSUMED - 122,076 features



An Example Model
(category “grain”)

Word	Beta		Word	Beta
corn	29.78		formal	-1.15
wheat	20.56		holder	-1.43
rice	11.33		hungarian	-6.15
sindt	10.56		rubber	-7.12
madagascar	6.83		special	-7.25
import	6.79		…	…
grain	6.77		beet	-13.24
contract	3.08		rockwood	-13.61

Text Sequence Modeling

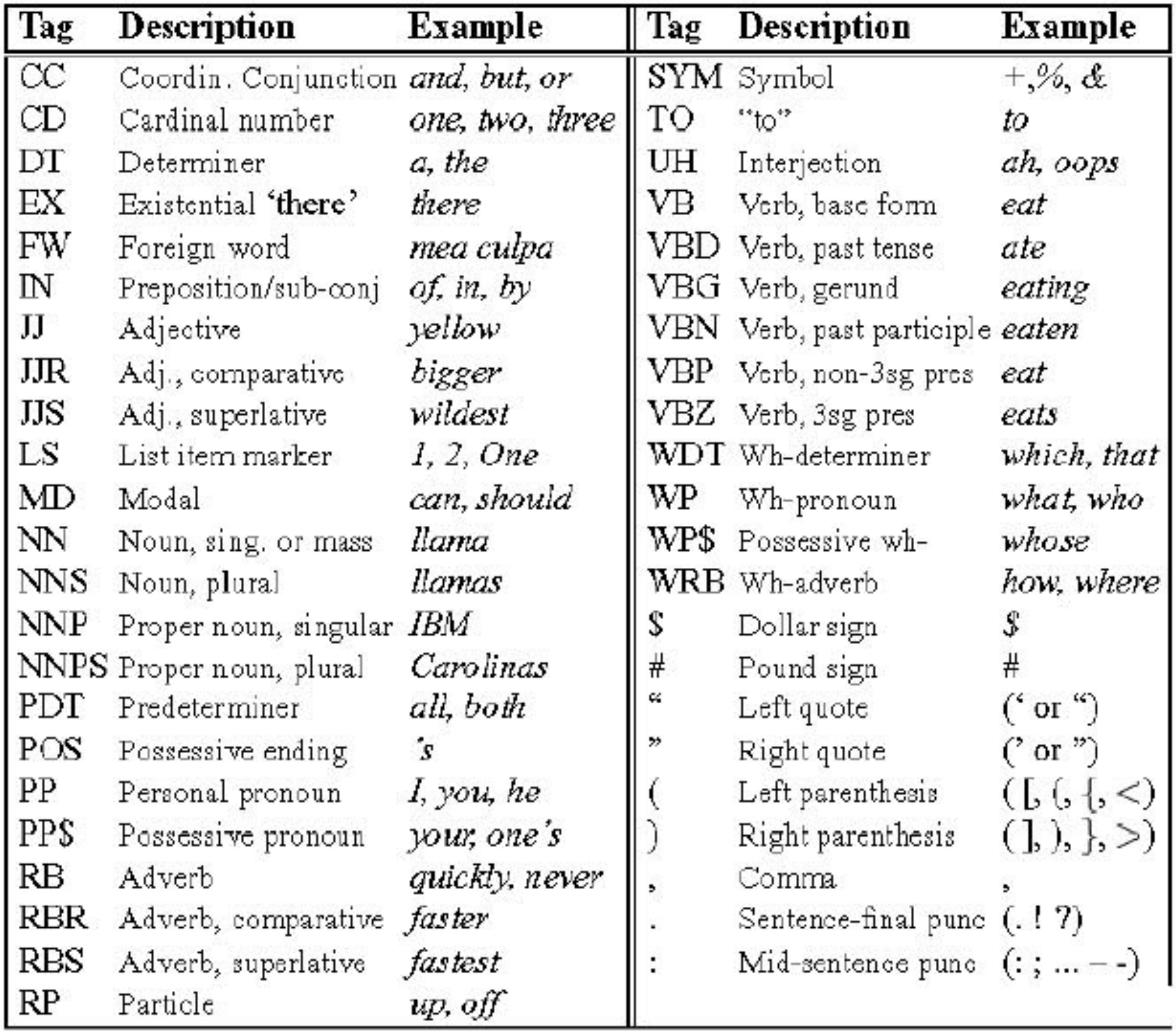
Introduction

- Textual data comprise sequences of words: "The quick brown fox…"
- Many tasks can put this sequence information to good use:
  - Part of speech tagging
  - Named entity extraction
  - Text chunking
  - Author identification

Part-of-Speech Tagging

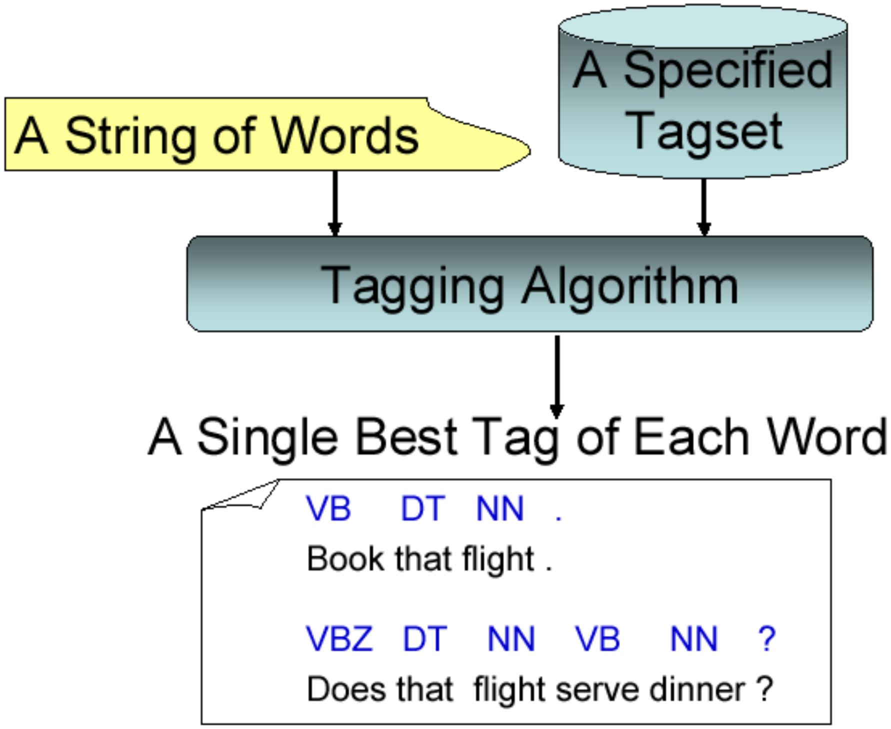
- Assign grammatical tags to words
- Basic task in the analysis of natural language data
- Phrase identification, entity extraction, etc.
- Ambiguity: “tag” could be a noun or a verb
- “a tag is a part-of-speech label” – context resolves the ambiguity

The Penn Treebank POS Tag Set



Tag	Description	Example	Tag	Description	Example
CC	Coordin. Conjunction	*and, but, or*	SYM	Symbol	+,%, &
CD	Cardinal number	*one, two, three*	TO	"to"	*to*
DT	Determiner	*a, the*	UH	Interjection	*ah, oops*
EX	Existential 'there'	*there*	VB	Verb, base form	*eat*
FW	Foreign word	*mea culpa*	VBD	Verb, past tense	*ate*
IN	Preposition/sub-conj	*of, in, by*	VBG	Verb, gerund	*eating*
JJ	Adjective	*yellow*	VBN	Verb, past participle	*eaten*
JJR	Adj., comparative	*bigger*	VBP	Verb, non-3sg pres	*eat*
JJS	Adj., superlative	*wildest*	VBZ	Verb, 3sg pres	*eats*
LS	List item marker	*1, 2, One*	WDT	Wh-determiner	*which, that*
MD	Modal	*can, should*	WP	Wh-pronoun	*what, who*
NN	Noun, sing. or mass	*llama*	WP$	Possessive wh-	*whose*
NNS	Noun, plural	*llamas*	WRB	Wh-adverb	*how, where*
NNP	Proper noun, singular	*IBM*	$	Dollar sign	$
NNPS	Proper noun, plural	*Carolinas*	#	Pound sign	#
PDT	Predeterminer	*all, both*	“	Left quote	(‘ or “)
POS	Possessive ending	*'s*	”	Right quote	(’ or ”)
PP	Personal pronoun	*I, you, he*	(	Left parenthesis	([, (, {, <)
PP$	Possessive pronoun	*your, one's*	)	Right parenthesis	(],), }, >)
RB	Adverb	*quickly, never*	,	Comma	,
RBR	Adverb, comparative	*faster*	.	Sentence-final punc	(. ! ?)
RBS	Adverb, superlative	*fastest*	:	Mid-sentence punc	(: ; ... – -)
RP	Particle	*up, off*			

POS Tagging Process



A String of Words

A Specified Tagset

Tagging Algorithm

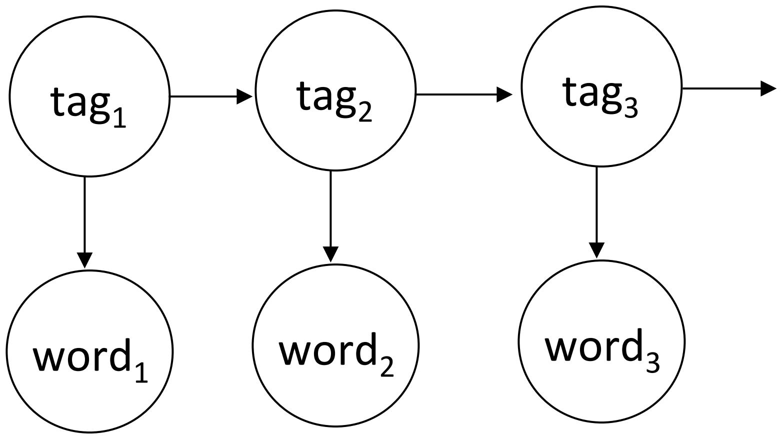
A Single Best Tag of Each Word

VB DT NN .
Book that flight .

VBZ DT NN VB NN ?
Does that flight serve dinner ?

POS Tagging Algorithms

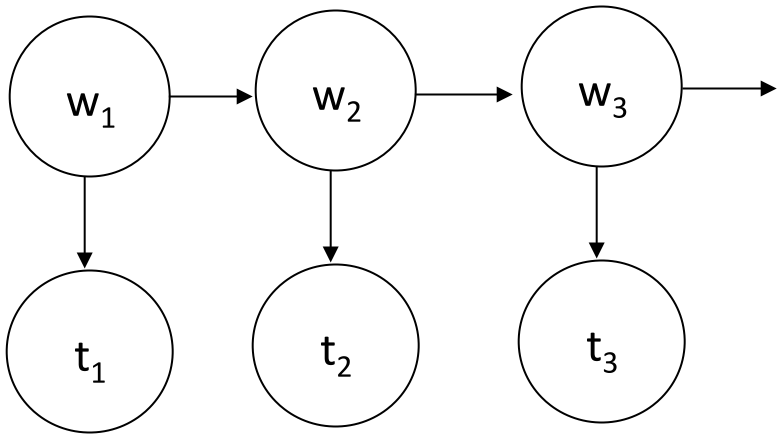
- Rule-based taggers: large numbers of hand-crafted rules
- Probabilistic tagger: used a tagged corpus to train some sort of model, e.g. HMM.



The Brown Corpus

- Comprises about 1 million English words
- HMM's first used for tagging on the Brown Corpus
- 1967. Somewhat dated now.
- British National Corpus has 100 million words

Simple Charniak Model



- What about words that have never been seen before?
- Clever tricks for smoothing the number of parameters (aka priors)

some details…

$$P(t^i \mid w^j) \overset{\text{est}}{=} \lambda_1(w^j)\frac{C(t^i, w^j)}{C(w^j)} + \lambda_2(w^j)\frac{\mathrm{C}_n(t^i)}{\mathrm{C}_n()}$$

$C(t^i, w^j)$ number of times word *j* appears with tag *i*

$C(w^j)$ number of times word *j* appears

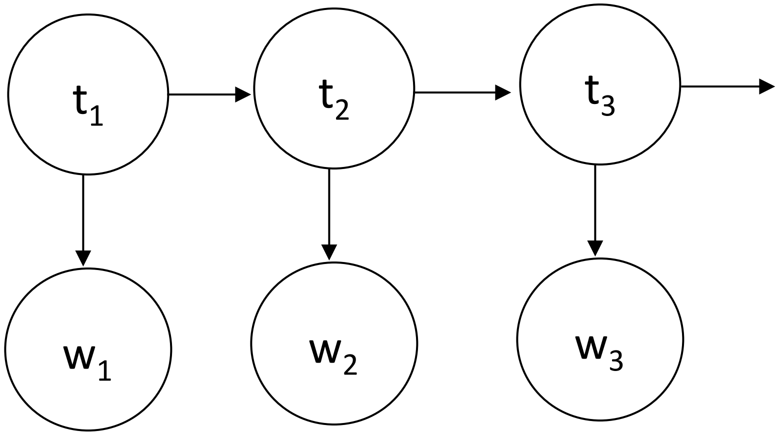
$\mathrm{C}_n(t^i)$ number of times a word that had never been seen with tag *i* gets tag *i*

$\mathrm{C}_n()$ number of such occurrences in total

$$\lambda_1(w^j) = \begin{cases} 1 & \text{if } C(w^j) \geq 1 \\ 0 & \text{otherwise.} \end{cases}$$

Test data accuracy on Brown Corpus = 91.51%

HMM



$$\mathcal{T}(w_{\mathbf{1},n}) = \arg\max_{t_{\mathbf{1},n}} \prod_{i=1}^{n} P(t_i \mid t_{i-1}) P(w_i \mid t_i)$$

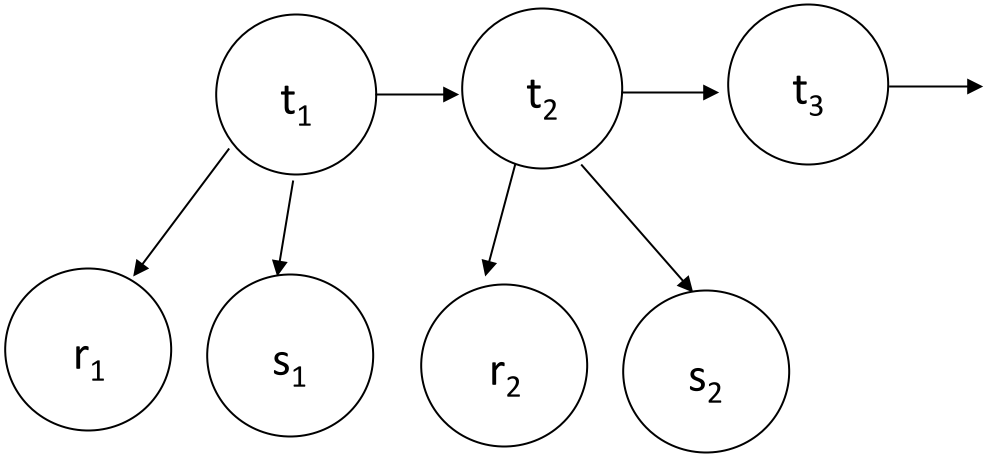
$$= \quad \arg\max_{t_{1,n}} \prod_{i=1}^{n} P(t_i \mid t_{i-1}) \frac{P(t_i \mid w_i)}{P(t_i)}$$

$$P(t_i \mid t_{i-1}) \overset{\text{est}}{=} (1-\epsilon) \frac{C(t_{i-1}, t_i)}{C(t_{i-1})} + \epsilon$$

Brown test set accuracy = 95.97%

Morphological Features

- Knowledge that “quickly” ends in “ly” should help identify the word as an adverb
- “randomizing” -> “ing”
- Split each word into a root (“quick”) and a suffix (“ly”)



Morphological Features

- Typical morphological analyzers produce multiple possible splits
- “Gastroenteritis” ???

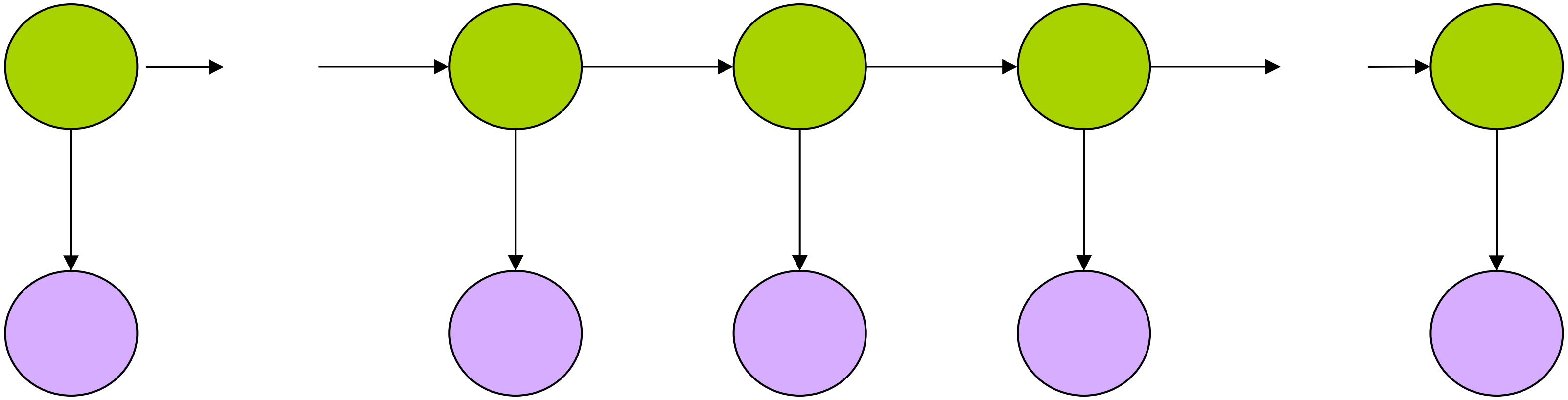
$$\mathcal{T}(w_{1,n}) = \arg\max_{t_{1,n}} \sum_{r_{1,n},s_{1,n}} \prod_{i=1}^{n} P(t_i \mid t_{i-1}) P(s_i \mid t_i) P(r_i \mid t_i) \quad (35)$$

$$= \arg\max_{t_{1,n}} \sum_{r_{1,n},s_{1,n}} \prod_{i=1}^{n} P(t_i \mid t_{i-1}) P(s_i \mid t_i) \frac{P(r_i)P(t_i \mid r_i)}{P(t_i)} \quad (36)$$

$$P(r^j) \overset{\text{est}}{=} \frac{5^{|r^j|}}{\mid r^j \mid!} e^{-5} \prod_{k=1}^{|r^j|} P(l_i \mid l_{i-1})$$

- Achieves 96.45% on the Brown Corpus

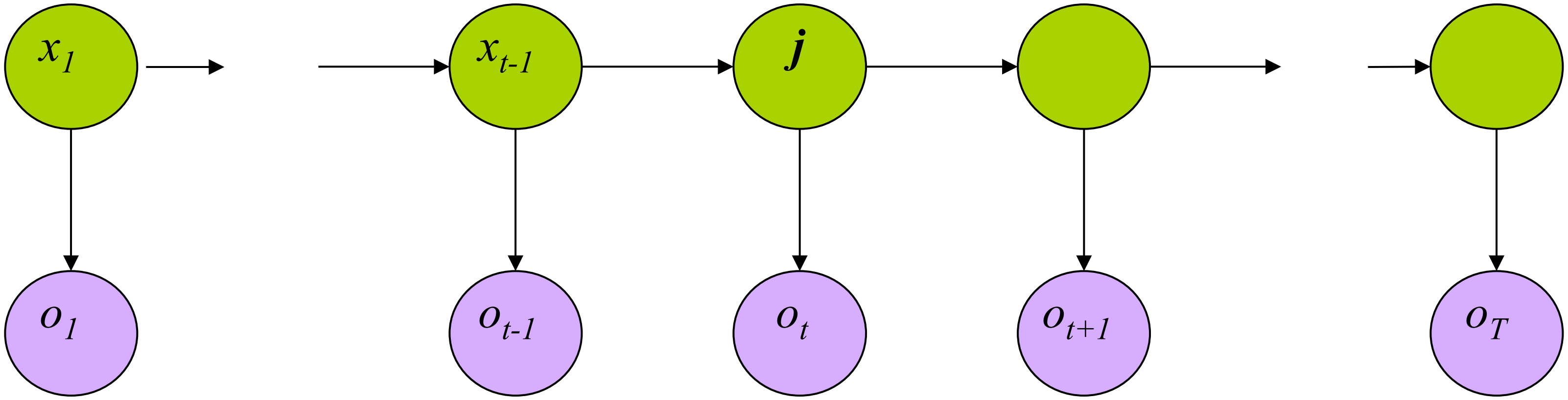
Inference in an HMM



- Compute the probability of a given observation sequence
- Given an observation sequence, compute the most likely hidden state sequence
- Given an observation sequence and set of possible models, which model most closely fits the data?

David Blei

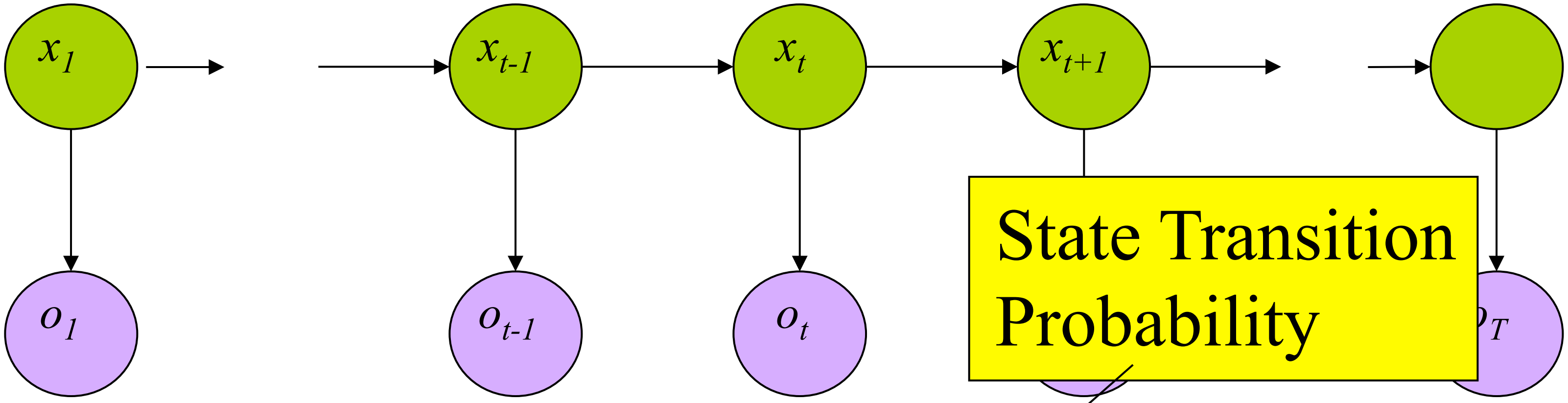
Viterbi Algorithm



$$\delta_j(t) = \max_{x_1 \ldots x_{t-1}} P(x_1 \ldots x_{t-1}, o_1 \ldots o_{t-1}, x_t = j, o_t)$$

The state sequence which maximizes the probability of seeing the observations to time t-1, landing in state j, and seeing the observation at time t

Viterbi Algorithm



State Transition Probability

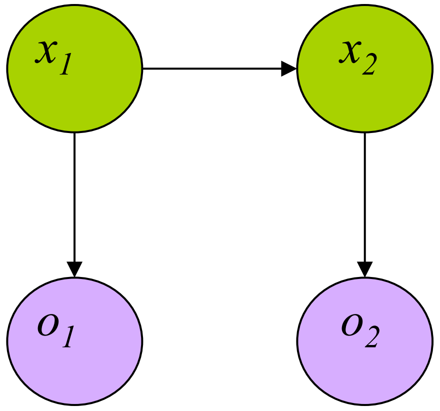
$$\delta_j(t) = \max_{x_1 \ldots x_{t-1}} P(x_1 \ldots x_{t-1}, o_1 \ldots o_{t-1}, x_t =$$

“Emission” Probability

$$\delta_j(t+1) = \max_i \delta_i(t) a_{ij} b_{jo_{t+1}}$$

Recursive Computation

Viterbi Small Example



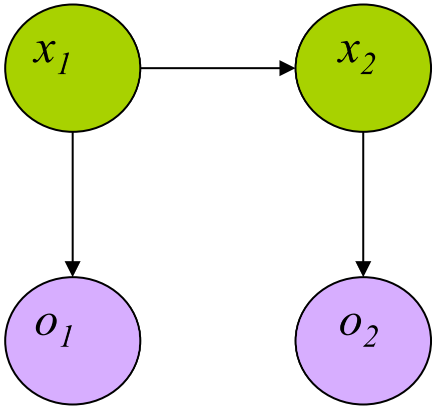
$\Pr(x_1\text{=T}) = 0.2$
$\Pr(x_2\text{=T}|x_1\text{=T}) = 0.7$
$\Pr(x_2\text{=T}|x_1\text{=F}) = 0.1$
$\Pr(o\text{=T}|x\text{=T}) = 0.4$
$\Pr(o\text{=T}|x\text{=F}) = 0.9$
o_1=T; o_2=F

Brute Force

$\Pr(x_1\text{=T},x_2\text{=T}, o_1\text{=T},o_2\text{=F}) = 0.2 \text{ x } 0.4 \text{ x } 0.7 \text{ x } 0.6 = 0.0336$
$\Pr(x_1\text{=T},x_2\text{=F}, o_1\text{=T},o_2\text{=F}) = 0.2 \text{ x } 0.4 \text{ x } 0.3 \text{ x } 0.1 = 0.0024$
$\Pr(x_1\text{=F},x_2\text{=T}, o_1\text{=T},o_2\text{=F}) = 0.8 \text{ x } 0.9 \text{ x } 0.1 \text{ x } 0.6 = 0.0432$
$\mathbf{\Pr(x_1\textbf{=F},x_2\textbf{=F}, o_1\textbf{=T},o_2\textbf{=F}) = 0.8 \text{ x } 0.9 \text{ x } 0.9 \text{ x } 0.1 = 0.0648}$

$\Pr(X_1,X_2 \mid o_1\text{=T},o_2\text{=F}) \propto \Pr(X_1,X_2 , o_1\text{=T},o_2\text{=F})$

Viterbi Small Example



$$\hat{X}_2 = \arg\max_j \delta_j(2)$$

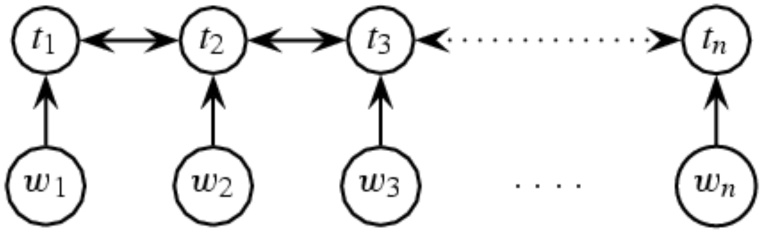
$$\delta_j(2) = \max_i \delta_i(1) a_{ij} b_{jo_2}$$

$$\delta_T(1) = \Pr(x_1 = T)\Pr(o_1 = T \mid x_1 = T) = 0.2 \times 0.4 = 0.08$$
$$\delta_F(1) = \Pr(x_1 = F)\Pr(o_1 = T \mid x_1 = F) = 0.8 \times 0.9 = 0.72$$

$$\delta_T(2) = \max\left(\delta_F(1) \times \Pr(x_2 = T \mid x_1 = F)\Pr(o_2 = F \mid x_2 = T), \delta_T(1) \times \Pr(x_2 = T \mid x_1 = T)\Pr(o_2 = F \mid x_2 = T)\right)$$
$$= \max(\underline{0.72 \times 0.1 \times 0.6}, 0.08 \times 0.7 \times 0.6) = 0.0432$$
$$\delta_F(2) = \max\left(\delta_F(1) \times \Pr(x_2 = F \mid x_1 = F)\Pr(o_2 = F \mid x_2 = F), \delta_T(1) \times \Pr(x_2 = F \mid x_1 = T)\Pr(o_2 = F \mid x_2 = F)\right)$$
$$= \max(\underline{0.72 \times 0.9 \times 0.1}, 0.0.08 \times 0.3 \times 0.1) = 0.0648$$

Other Developments

- Toutanova et al., 2003, use a “dependency network” and richer feature set

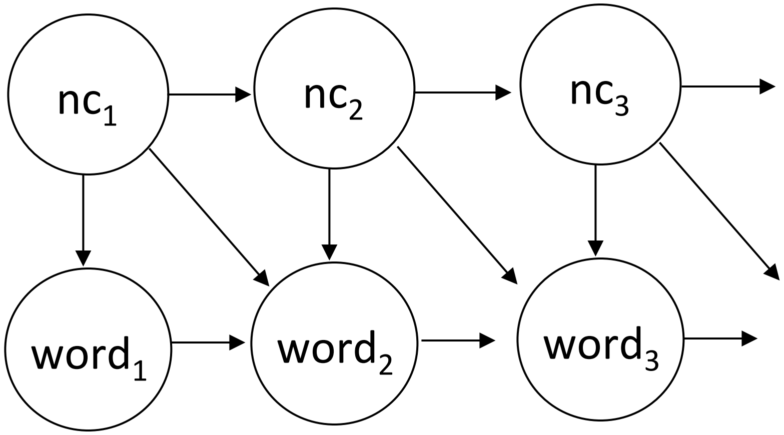


•Idea: using the “next” tag as well as the “previous” tag should improve tagging performance

Named-Entity Classification

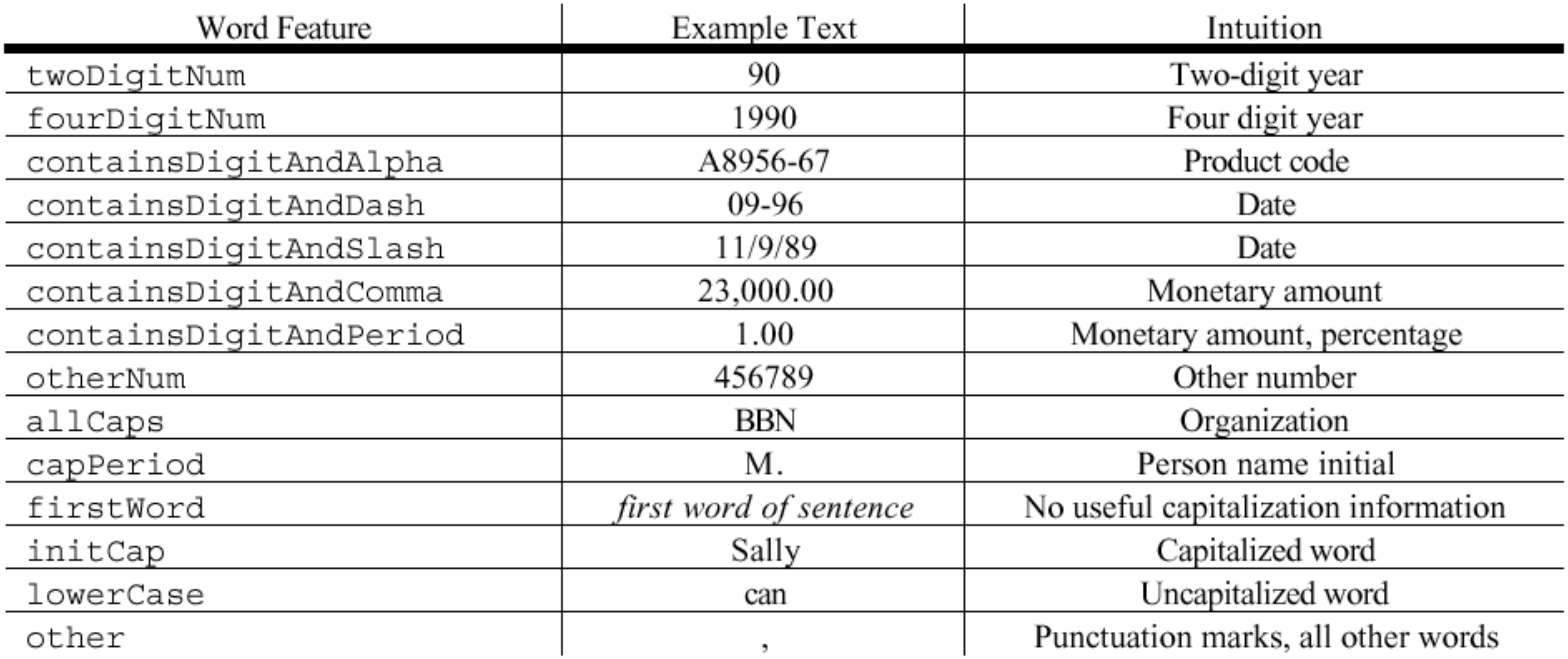
- "*Mrs. Frank*" is a person
- "*Steptoe and Johnson*" is a company
- "*Honduras*" is a location
- etc.

  - Bikel et al. (1998) from BBN "Nymble" statistical approach using HMMs



$$[w_i \mid w_{i-1}, nc_i, nc_{i-1}] = \begin{cases} [w_i \mid w_{i-1}, nc_i] & \text{if } nc_i = nc_{i-1} \\ [w_i \mid nc_i, nc_{i-1}] & \text{if } nc_i \neq nc_{i-1} \end{cases}$$

- “name classes”: Not-A-Name, Person, Location, etc.
- Smoothing for sparse training data + word features
- Training = 100,000 words from WSJ
- Accuracy = 93%
- 450,000 words → same accuracy



Word Feature	Example Text	Intuition
twoDigitNum	90	Two-digit year
fourDigitNum	1990	Four digit year
containsDigitAndAlpha	A8956-67	Product code
containsDigitAndDash	09-96	Date
containsDigitAndSlash	11/9/89	Date
containsDigitAndComma	23,000.00	Monetary amount
containsDigitAndPeriod	1.00	Monetary amount, percentage
otherNum	456789	Other number
allCaps	BBN	Organization
capPeriod	M.	Person name initial
firstWord	*first word of sentence*	No useful capitalization information
initCap	Sally	Capitalized word
lowerCase	can	Uncapitalized word
other	,	Punctuation marks, all other words

training-development-test