Text Classification:
An Advanced Tutorial

William W. Cohen
Machine Learning Department, CMU

Outline

- Part I: the basics
  - What is text classification? Why do it?
  - Representing text for classification
  - A simple, fast generative method
  - Some simple, fast discriminative methods
- Part II: advanced topics
  - Sentiment detection and subjectivity
  - Collective classification
  - Alternatives to bag-of-words

Text Classification: definition

- The classifier:
  - *Input*: a document x
  - *Output*: a predicted class y from some fixed set of labels $y_1, \ldots, y_K$
- The learner:
  - *Input:* a set of m hand-labeled documents $(x_1, y_1), \ldots, (x_m, y_m)$
  - *Output:* a learned classifier $f: x \rightarrow y$

Text Classification: Examples

- Classify news stories as *World, US, Business, SciTech, Sports, Entertainment, Health, Other*
- Add MeSH terms to Medline abstracts
  - e.g. "Conscious Sedation" [E03.250]
- Classify business names by industry.
- Classify student essays as *A,B,C,D,* or *F.*
- Classify email as *Spam, Other.*
- Classify email to tech staff as *Mac, Windows, ..., Other.*
- Classify pdf files as *ResearchPaper, Other*
- Classify documents as *WrittenByReagan, GhostWritten*
- Classify movie reviews as *Favorable,Unfavorable,Neutral.*
- Classify technical papers as *Interesting, Uninteresting.*
- Classify jokes as *Funny, NotFunny.*
- Classify web sites of companies by Standard Industrial Classification (SIC) code.

Text Classification: Examples

- Best-studied benchmark: *Reuters-21578* newswire stories
  - 9603 train, 3299 test documents, 80-100 words each, 93 classes

ARGENTINE 1986/87 GRAIN/OILSEED REGISTRATIONS
BUENOS AIRES, Feb 26
Argentine grain board figures show crop registrations of grains, oilseeds and their products to February 11, in thousands of tonnes, showing those for future shipments month, 1986/87 total and 1985/86 total to February 12, 1986, in brackets:

- Bread wheat prev 1,655.8, Feb 872.0, March 164.6, total 2,692.4 (4,161.0).
- Maize Mar 48.0, total 48.0 (nil).
- Sorghum nil (nil)
- Oilseed export registrations were:
- Sunflowerseed total 15.0 (7.9)
- Soybean May 20.0, total 20.0 (nil)

The board also detailed export registrations for subproducts, as follows....

→ Categories: **grain**, **wheat** (of 93 binary choices)

Representing text for classification

f(

ARGENTINE 1986/87 GRAIN/OILSEED REGISTRATIONS
BUENOS AIRES, Feb 26

Argentine grain board figures show crop registrations of grains, oilseeds and their products to February 11, in thousands of tonnes, showing those for future shipments month, 1986/87 total and 1985/86 total to February 12, 1986, in brackets:

- Bread wheat prev 1,655.8, Feb 872.0, March 164.6, total 2,692.4 (4,161.0).
- Maize Mar 48.0, total 48.0 (nil).
- Sorghum nil (nil)
- Oilseed export registrations were:
- Sunflowerseed total 15.0 (7.9)
- Soybean May 20.0, total 20.0 (nil)

The board also detailed export registrations for subproducts, as follows....

)=y

?

simplest useful

What is the ~~**best**~~ representation for the document *x* being classified?

Representing text: a list of words

f(

ARGENTINE 1986/87 GRAIN/OILSEED REGISTRATIONS
BUENOS AIRES, Feb 26
Argentine grain board figures show crop registrations of grains, oilseeds and their products to February 11, in thousands of tonnes, showing those for future shipments month, 1986/87 total and 1985/86 total to February 12, 1986, in brackets:

- Bread wheat prev 1,655.8, Feb 872.0, March 164.6, total 2,692.4 (4,161.0).
- Maize Mar 48.0, total 48.0 (nil).
- Sorghum nil (nil)
- Oilseed export registrations were:
- Sunflowerseed total 15.0 (7.9)
- Soybean May 20.0, total 20.0 (nil)

The board also detailed export registrations for subproducts, as follows....

)=y

f(

(argentine, 1986, 1987, grain, oilseed, registrations, buenos, aires, feb, 26, argentine, grain, board, figures, show, crop, registrations, ~~of,~~ grains, oilseeds, ~~and,~~ ~~their,~~ products, ~~to,~~ february, 11, ~~in,~~ …

)=y

Common refinements: remove stopwords, stemming, collapsing multiple occurrences of words into one….

Text Classification with Naive Bayes

- Represent document *x* as list of words *w1,w2,…*
- For each y, build a probabilistic model Pr(X|Y=y) of "documents" in class y
  - Pr(X=*{argentine,grain...}*|Y=*wheat*) = ...
  - Pr(X=*{stocks,rose,in,heavy,...}*|Y=*nonWheat*) = ...
- To classify, find the *y* which was most likely to *generate x—i.e.*, which gives *x* the best score according to Pr(x|y)
  - $f(x) = \mathrm{argmax}_y \Pr(x|y) * \Pr(y)$

Text Classification with Naive Bayes

- How to estimate Pr(X|Y) ?
- *Simplest useful* process to generate a bag of words:
  - pick word 1 according to Pr(W|Y)
  - repeat for word 2, 3, ...
  - each word is generated *independently* of the others (which is clearly not true) but means

$$\Pr(w_1,\ldots,w_n \mid Y=y) = \prod_{i=1}^{n} \underbrace{\Pr(w_i \mid Y=y)}$$

How to estimate Pr(W|Y)?

Text Classification with Naive Bayes

- How to estimate Pr(X|Y) ?

$$\Pr(w_1,\ldots,w_n \mid Y=y) = \prod_{i=1}^{n} \underbrace{\Pr(w_i \mid Y=y)}$$

Estimate *Pr(w|y)* by looking at the data...

$$\Pr(W=w \mid Y=y) = \frac{\text{count}(W=w \text{ and } Y=y)}{\text{count}(Y=y)}$$

This gives score of zero if x contains a brand-new word w_{new}

Text Classification with Naive Bayes

- How to estimate Pr(X|Y) ?

$$\Pr(w_1,\ldots,w_n \mid Y = y) = \prod_{i=1}^{n} \underbrace{\Pr(w_i \mid Y = y)}$$

... and also **imagine** *m* examples with *Pr(w|y)=p*

$$\Pr(W = w \mid Y = y) = \frac{\mathrm{count}(W = w \text{ and } Y = y) + mp}{\mathrm{count}(Y = y) + m}$$

Terms:

- This Pr(W|Y) is a *multinomial distribution*
- This use of *m* and *p* is a *Dirichlet prior* for the multinomial

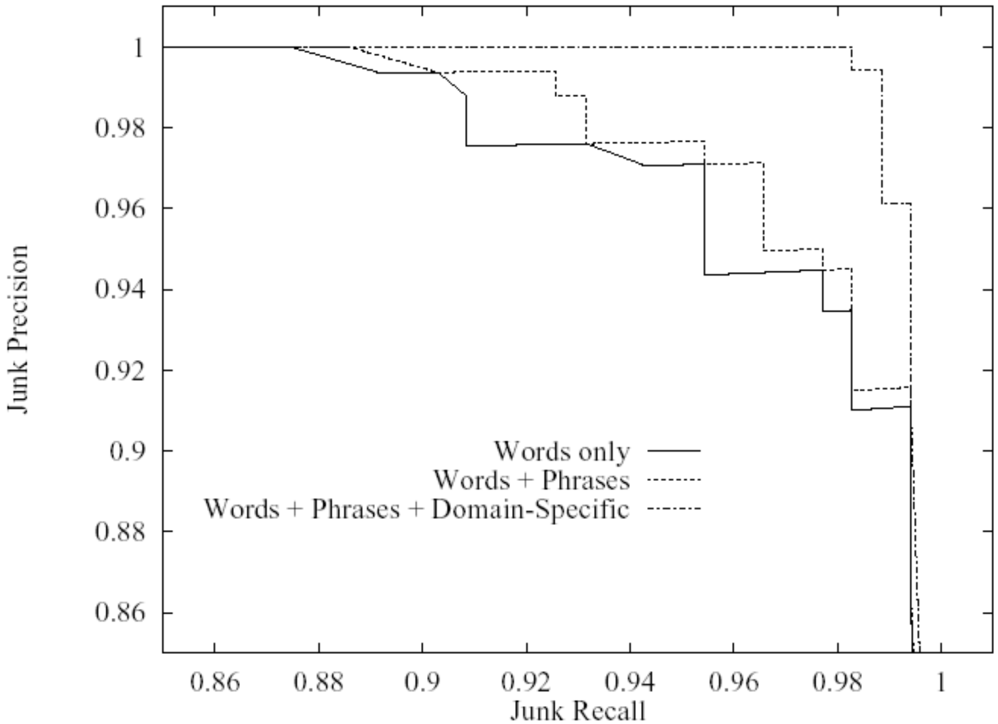
Text Classification with Naive Bayes

- Putting this together:
  - for each document x_i with label y_i
    - for each word w_{ij} in x_i
      - count[w_{ij}][y_i]++
      - count[y_i]++
      - count++
  - to classify a new $x=w_1...w_n$, pick y with top *score*:

$$score(y, w_1...w_k) = \lg \frac{\text{count}[y]}{\text{count}} + \sum_{i=1}^{n} \lg \frac{\text{count}[w_i][y] + 0.5}{\text{count}[y] + 1}$$

key point: we only need counts for words that actually appear in *x*

Naïve Bayes for SPAM filtering
(Sahami et al, 1998)



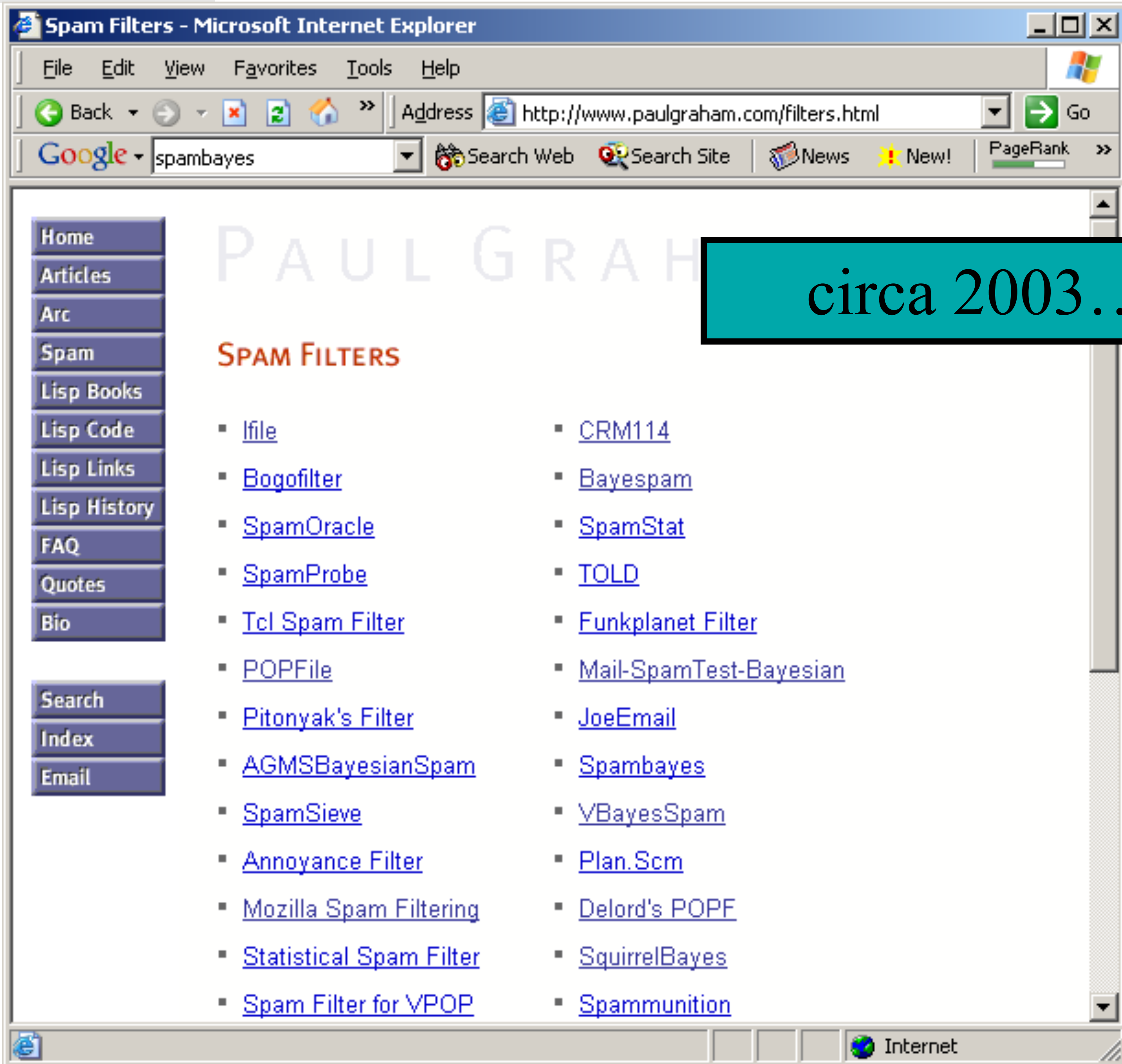
Used bag of words,
+ special phrases
("FREE!") and +
special features
("from *.edu", …)

Terms: *precision, recall*

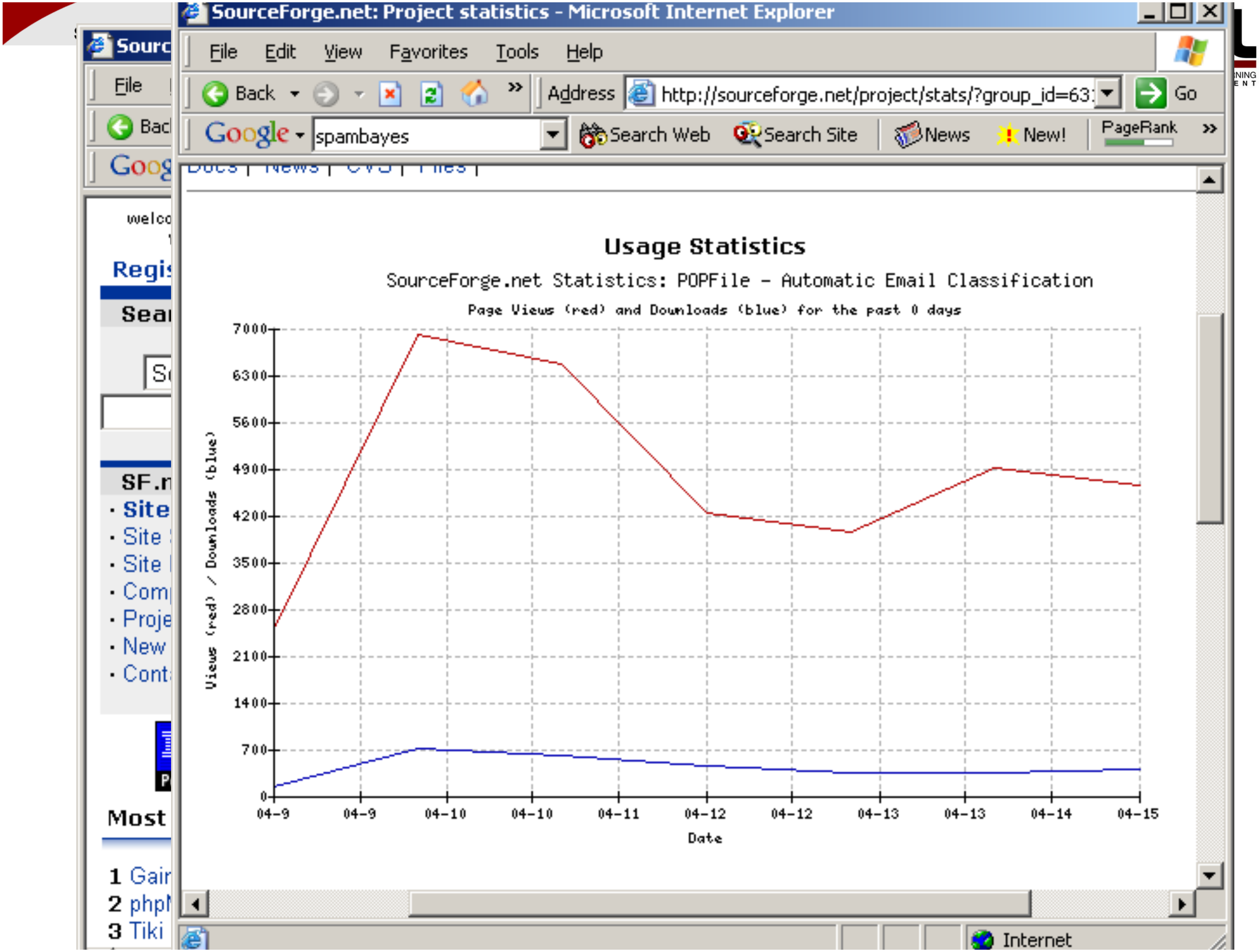
	Classified Junk	Classified Legitimate	Total
Actually Junk	36 (92.0% precision)	9	45
Actually Legitimate	3	174 (95.0% precision)	177
Total	39	183	222

Carnegie Mellon
SCHOOL OF COMPUTER SCIENCE

MACHINE LEARNING DEPARTMENT



circa 2003…


SourceForge.net: Project statistics - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://sourceforge.net/project/stats/?group_id=63
Go
Google
spambayes
Search Web
Search Site
News
New!
PageRank
Usage Statistics
SourceForge.net Statistics: POPFile - Automatic Email Classification
Page Views (red) and Downloads (blue) for the past 0 days
Views (red) / Downloads (blue)
7000
6300
5600
4900
4200
3500
2800
2100
1400
700
0
04-9
04-9
04-10
04-10
04-11
04-12
04-12
04-13
04-13
04-14
04-15
Date
Internet

Naive Bayes Summary

- Pros:
  - Very fast and easy-to-implement
  - Well-understood formally & experimentally
    - see "Naive (Bayes) at Forty", Lewis, ECML98
- Cons:
  - Seldom gives the very best performance
  - "Probabilities" *Pr(y|x)* are not accurate
    - e.g., Pr(y|x) decreases with length of *x*
    - Probabilities tend to be close to zero or one

Outline

- Part I: the basics
  - What is text classification? Why do it?
  - Representing text for classification
  - A simple, fast generative method
  - Some simple, fast discriminative methods
- Part II: advanced topics
  - Sentiment detection and subjectivity
  - Collective classification
  - Alternatives to bag-of-words

Representing text: a list of words

f(

ARGENTINE 1986/87 GRAIN/OILSEED REGISTRATIONS
BUENOS AIRES, Feb 26
Argentine grain board figures show crop registrations of grains, oilseeds and their products to February 11, in thousands of tonnes, showing those for future shipments month, 1986/87 total and 1985/86 total to February 12, 1986, in brackets:

- Bread wheat prev 1,655.8, Feb 872.0, March 164.6, total 2,692.4 (4,161.0).
- Maize Mar 48.0, total 48.0 (nil).
- Sorghum nil (nil)
- Oilseed export registrations were:
- Sunflowerseed total 15.0 (7.9)
- Soybean May 20.0, total 20.0 (nil)

The board also detailed export registrations for subproducts, as follows....

)=y

f(

(argentine, 1986, 1987, grain, oilseed, registrations, buenos, aires, feb, 26, argentine, grain, board, figures, show, crop, registrations, ~~of~~, grains, oilseeds, ~~and~~, ~~their~~, products, ~~to~~, february, 11, ~~in~~, …

)=y

Common refinements: remove stopwords, stemming, collapsing multiple occurrences of words into one….

Representing text: a bag of words

ARGENTINE 1986/87 **GRAIN/OILSEED** REGISTRATIONS
BUENOS AIRES, Feb 26

Argentine **grain** board figures show crop registrations of **grains**, **oilseeds** and their products to February 11, in thousands of **tonnes**, showing those for future **shipments** month, 1986/87 **total** and 1985/86 **total** to February 12, 1986, in brackets:

- Bread **wheat** prev 1,655.8, Feb 872.0, March 164.6, **total** 2,692.4 (4,161.0).
- **Maize** Mar 48.0, total 48.0 (nil).
- **Sorghum** nil (nil)
- **Oilseed** export registrations were:
- **Sunflowerseed** total 15.0 (7.9)
- **Soybean** May 20.0, total 20.0 (nil)

The board also detailed export registrations for subproducts, as follows....

word	*freq*
grain(s)	3
oilseed(s)	2
total	3
wheat	1
maize	1
soybean	1
tonnes	1
...	...

If the order of words doesn't matter, **x** can be a *vector* of word *frequencies*.

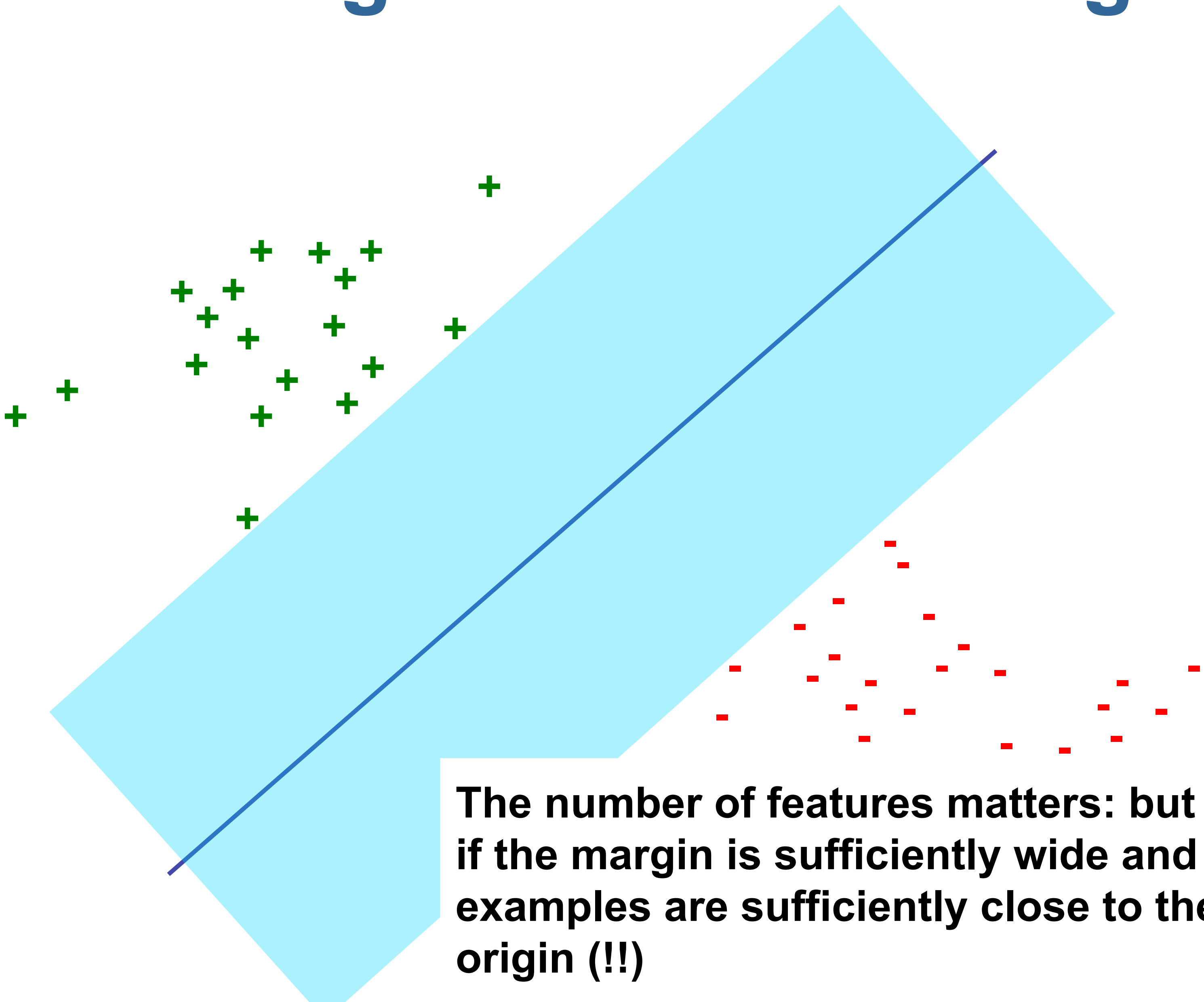
Categories: **grain**, **wheat**

"Bag of words": a long sparse vector **x**=(,…,f_i,….) where f_i is the frequency of the i-th word in the vocabulary

The Curse of Dimensionality

- First serious experimental look at TC:
  - Lewis's 1992 thesis
    - Reuters-21578 is from this, cleaned up *circa* 1996-7
  - Compare to Fisher's linear discriminant 1936 (*iris* data)
  - Why did it take so long to look at text classification?
- Scale:
  - Typical text categorization problem: *TREC-AP* headlines (Cohen&Singer,2000): 319,000+ documents, 67,000+ words, 3,647,000+ word 4-grams used as features.
- *How can you learn with so many features?*
  - For efficiency (time & memory), use *sparse* vectors.
  - Use simple classifiers (linear or loglinear)
  - Rely on *wide margins*.

Margin-based Learning



The number of features matters: but not if the margin is sufficiently wide and examples are sufficiently close to the origin (!!)

The Voted Perceptron

[Freund & Schapire, 1998]

- Assume y=±1
- Start with $v_1 = (0,...,0)$
- For example (x_i, y_i):
  - $y' = sign(v_k \cdot x_i)$
  - if y' is correct, $c_k{+}{+}$;
  - if y' is not correct:
    - $v_{k+1} = v_k + y_i x_k$
    - $k = k+1$
    - $c_{k+1} = 1$
- Classify by voting all v_k's predictions, weighted by c_k

An amazing fact: **if**
- for all i, $||x_i||<R$,
- there is some u so that $||u||=1$ and for all i, $y_i*(u.x)>\delta$ **then** the voted perceptron makes few mistakes: less than $(R/\delta)^2$

For text with binary features: $||x_i|| <R$ means not too many words.

And $yi*(u.x)>\delta$ means the margin is at least δ

The Voted Perceptron: Proof

Theorem: **if**
- for all i, $||x_i||<R$,
- there is some u so that $||u||=1$ and for all i, $y_i*(u.x_i)>\delta$ **then** the perceptron makes few mistakes: less than $(R/\delta)^2$

1) "Mistake" implies $v_{k+1} = v_k + y_i x_i$

$\Rightarrow$ $u.v_{k+1} = u(v_k + y_i x_k)$

$\Rightarrow$ $u.v_{k+1} = u.v_k + u y_i x_k$

$\Rightarrow$ $u.v_{k+1} > u.v_k + \delta$

So u.v, and hence v, **grows** by at least δ: $v_{k+1}.u>k\,\delta$

2) "Mistake" also implies $y_i(v_k.x_i) < 0$

$\Rightarrow$ $||v_{k+1}||^2 = ||v_k + y_i x_i||^2$

$\Rightarrow$ $||v_{k+1}||^2 = ||v_k|| + 2y_i(v_k.x_i) + ||x_i||^2$

$\Rightarrow$ $||v_{k+1}||^2 < ||v_k|| + 2y_i(v_k.x_i) + R^2$

$\Rightarrow$ $||v_{k+1}||^2 < ||v_k|| + R^2$

So v cannot grow too much with each mistake: $||v_{k+1}||^2 < k\,R^2$

Two opposing forces:

- $||v_k||$ is squeezed between k δ and k^{-2}R
- this means that $k^{-2}R < k\,\delta$, which bounds k.

Lessons of the Voted Perceptron

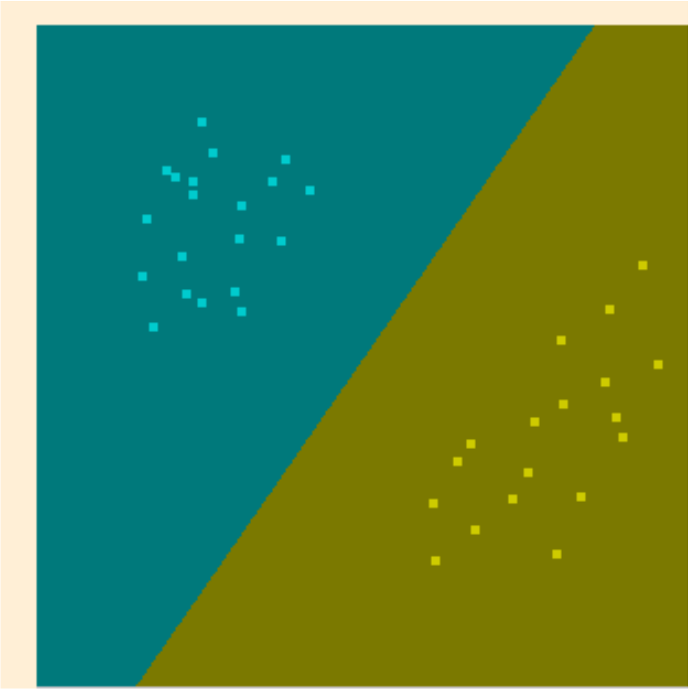
- VP shows that you can make **few mistakes** in **incrementally** learning as you pass over the data, if the examples *x* are **small** (bounded by *R*), some *u* exists that is **small** (unit norm) and has large **margin**.
- Why not look for this *u* directly?

Support vector machines:

• find *u* to minimize ||u||, subject to some fixed margin δ, or

• find *u* to maximize δ, relative to a fixed bound on ||u||.

• quadratic optimization methods



More on Support Vectors for Text

- Facts about support vector machines:
  - the "support vectors" are the x_i's that touch the margin.
  - the classifier *sign(u.x)* can be written

$$sign(\sum_i \alpha_i (x_i \cdot x))$$

  where the x_i's are the support vectors.
  - the inner products $x_i.x$ can be replaced with variant "kernel functions"
  - support vector machines often give very good results on topical text classification.

Support Vector Machine Results

[Joacchim ECML 1998]

	Bayes	Rocchio	C4.5	k-NN	SVM (poly) degree $d =$					SVM (rbf) width $\gamma =$			
					1	2	3	4	5	0.6	0.8	1.0	1.2
earn	95.9	96.1	96.1	97.3	98.2	98.4	**98.5**	98.4	98.3	**98.5**	98.5	98.4	98.3
acq	91.5	92.1	85.3	92.0	92.6	94.6	**95.2**	95.2	95.3	95.0	95.3	95.3	**95.4**
money-fx	62.9	67.6	69.4	78.2	66.9	72.5	75.4	74.9	**76.2**	74.0	75.4	**76.3**	75.9
grain	72.5	79.5	89.1	82.2	91.3	93.1	**92.4**	91.3	89.9	**93.1**	91.9	91.9	90.6
crude	81.0	81.5	75.5	85.7	86.0	87.3	88.6	**88.9**	87.8	**88.9**	89.0	88.9	88.2
trade	50.0	77.4	59.2	77.4	69.2	75.5	76.6	77.3	**77.1**	76.9	78.0	**77.8**	76.8
interest	58.0	72.5	49.1	74.0	69.8	63.3	67.9	73.1	**76.2**	74.4	75.0	**76.2**	76.1
ship	78.7	83.1	80.9	79.2	82.0	85.4	86.0	**86.5**	86.0	**85.4**	86.5	87.6	87.1
wheat	60.6	79.4	85.5	76.6	83.1	84.5	85.2	**85.9**	83.8	**85.2**	85.9	85.9	85.9
corn	47.3	62.2	87.7	77.9	86.0	86.5	85.3	**85.7**	83.9	**85.1**	85.7	85.7	84.5
microavg.	**72.0**	**79.9**	**79.4**	**82.3**	84.2	85.1	85.9	86.2	85.9	86.4	86.5	86.3	86.2
					combined: **86.0**					combined: **86.4**			

TF-IDF Representation

- The results above use a particular way to represent documents: *bag of words* with TFIDF weighting
  - "Bag of words": a long sparse vector $\mathbf{x}=(,\ldots,f_i,\ldots.)$ where f_i is the "weight" of the i-th word in the vocabulary
  - for word *w* that appears in *DF(w)* docs out of *N* in a collection, and appears *TF(w)* times in the doc being represented use weight:

$$f_{i(w)} = \log(TF(w)+1)\times\log\frac{N}{DF(w)}$$

  - also normalize all vector lengths (||x||) to 1

TF-IDF Representation

- TF-IDF representation is an old trick from the information retrieval community, and often improves performance of other algorithms:
  - Yang: extensive experiments with K-NN on TFIDF
    - Given **x** find *K* closest neighbors ($\mathbf{z}_1$,y_1) …, ($\mathbf{z}_K$,y_K)
    - Predict *y:*

$$\arg\max_y \sum_{(\mathbf{z},y'):y'=y} (\mathbf{x}\cdot\mathbf{z})$$

    - Implementation: use a TFIDF-based search engine to find neighbors
  - Rocchio's algorithm: classify using distance to centroids

$$sign(x\cdot w) \text{ where } w=\alpha\sum_{(\mathbf{z},+)}\mathbf{z}-\beta\sum_{(\mathbf{z},-)}\mathbf{z}$$

Support Vector Machine Results

[Joacchim ECML 1998]

					SVM (poly) degree $d =$					SVM (rbf) width $\gamma =$			
	Bayes	Rocchio	C4.5	k-NN	1	2	3	4	5	0.6	0.8	1.0	1.2
earn	95.9	96.1	96.1	97.3	98.2	98.4	**98.5**	98.4	98.3	**98.5**	98.5	98.4	98.3
acq	91.5	92.1	85.3	92.0	92.6	94.6	**95.2**	95.2	95.3	95.0	95.3	95.3	**95.4**
money-fx	62.9	67.6	69.4	78.2	66.9	72.5	75.4	74.9	**76.2**	74.0	75.4	**76.3**	75.9
grain	72.5	79.5	89.1	82.2	91.3	93.1	**92.4**	91.3	89.9	**93.1**	91.9	91.9	90.6
crude	81.0	81.5	75.5	85.7	86.0	87.3	88.6	**88.9**	87.8	**88.9**	89.0	88.9	88.2
trade	50.0	77.4	59.2	77.4	69.2	75.5	76.6	77.3	**77.1**	76.9	78.0	**77.8**	76.8
interest	58.0	72.5	49.1	74.0	69.8	63.3	67.9	73.1	**76.2**	74.4	75.0	**76.2**	76.1
ship	78.7	83.1	80.9	79.2	82.0	85.4	86.0	**86.5**	86.0	**85.4**	86.5	87.6	87.1
wheat	60.6	79.4	85.5	76.6	83.1	84.5	85.2	**85.9**	83.8	**85.2**	85.9	85.9	85.9
corn	47.3	62.2	87.7	77.9	86.0	86.5	85.3	**85.7**	83.9	**85.1**	85.7	85.7	84.5
microavg.	**72.0**	**79.9**	**79.4**	**82.3**	84.2	85.1	85.9	86.2	85.9	86.4	86.5	86.3	86.2
					combined: **86.0**					combined: **86.4**			

TF-IDF Representation

- TF-IDF representation is an old trick from the information retrieval community, and often improves performance of other algorithms:
  - Yang, CMU: extensive experiments with K-NN variants and linear least squares using TF-IDF representations
  - Rocchio's algorithm: classify using distance to centroid of documents from each class
  - Rennie et al: **Naive Bayes with TFIDF on "complement" of class**

	MNB	TWCNB	SVM	
Industry Sector	0.582	0.923	0.934	accuracy
20 Newsgroups	0.848	0.861	0.862	
Reuters (micro)	0.739	0.844	0.887	breakeven
Reuters (macro)	0.270	0.647	0.694	

Other Fast Discriminative Methods

[Carvalho & Cohen, KDD 2006]

Table 1: Mistake-Driven Online Learner.

1. Initialize $i = 0$, success counter $c_i = 0$, model w_0
2. For $t = 1, 2, ..., T$:
   (a) Receive new example x_t
   (b) Predict $\hat{y}_t = f(w_i, x_t)$, and receive true class y_t
   (c) If prediction was mistaken:
      i. Update model $w_i \rightarrow w_{i+1}$
      ii. $i = i + 1$
   (d) Else: $c_i = c_i + 1$

Table 2: Modified Balanced Winnow (MBW).

1. Initialize $i = 0$, counter $c_i = 0$, and models u_0 and v_0
2. For $t = 1, 2, ..., T$:
   (a) Receive new example x_t, and add "bias" feature.
   (b) Normalize x_t to 1.
   (c) Calculate $score = \langle x_t, u_i \rangle - \langle x_t, v_i \rangle - \theta_{th}$.
   (d) Receive true class y_t.
   (e) If prediction was mistaken, i.e., $(score \cdot y_t) \leq M$:
      i. Update models. For all feature j s.t. $x_t > 0$:

$$u_{i+1}^j = \begin{cases} u_i^j \cdot \alpha \cdot (1 + x_t^j) & \text{, if } y_t > 0 \\ u_i^j \cdot \beta \cdot (1 - x_t^j) & \text{, if } y_t < 0 \end{cases}$$

$$v_{i+1}^j = \begin{cases} v_i^j \cdot \beta \cdot (1 - x_t^j) & \text{, if } y_t > 0 \\ v_i^j \cdot \alpha \cdot (1 + x_t^j) & \text{, if } y_t < 0 \end{cases}$$

      ii. $i = i + 1$
   (f) Else: $c_i = c_i + 1$

Perceptron (w/o voting) is an example; another is Winnow.

There are many other examples.

• In practice they are usually *not* used on-line—instead one iterates over the data several times (epochs).

• What if you limit yourself to *one pass*? (which is all that Naïve Bayes needs!)

Other Fast Discriminative Methods

[Carvalho & Cohen, KDD 2006]

	SVM	v-P	MBW	v-MBW	NB
RequestAct	68.0	65.4	**76.7**	67.3	56.85
Spam	96.7	69.0	95.7	95.7	**97.4**
Scam	99.0	94.2	**99.9**	99.8	99.62
Reuters	96.7	96.3	95.9	**96.8**	85.52
20newsgroup	88.8	67.9	93.7	91.9	**94.42**
MovieReviews	**78.5**	71.4	75.1	77.1	71.85
Webmaster	**88.9**	88.5	88.6	86.6	77.38
Ads	80.5	58.0	**81.3**	78.2	52.5
Median F1	88.8	70.2	**91.1**	89.3	81.45
Avg. Rank	2.25	4.25	**2.12**	2.62	3.62
Signature	**80.3**	80.2	80.2	**80.3**	73.88
Reply-to	**94.8**	94.3	93.4	93.5	93.98
Adult	32.3	26.6	25.0	19.6	**41.0**
Congressional	**96.2**	95.7	94.2	95.9	91.7
Credit	**80.2**	59.5	72.1	79.6	66.78
WiscBreast	96.6	97.1	96.8	97.2	**98.2**
Nursery	**87.1**	86.8	57.0	69.6	84.4
Median F1	**87.1**	86.8	80.2	80.3	84.4
Avg. Rank	**1.71**	3.00	4.00	3.00	3.14

Sparse, high-dimensional TC problems

Dense, lower dimensional problems

Table 4: General Performance - F1 measure (%). NB=Naive Bayes, v-P= Voted Perceptron.

Other Fast Discriminative Methods

[Carvalho & Cohen, KDD 2006]

NLP Datasets	MBW	PW	BW	PA	ROMMA	v-MBW	v-PW	v-BW	v-PA	v-ROMMA
RequestAct	**76.7**	67.0^{**}	62.6^{**}	68.9^{*}	09.6^{**}	67.3^{**}	46.8^{**}	59.0^{**}	60.2^{**}	5.6^{**}
Spam	95.8	93.8^{**}	94.4	93.1^{**}	83.1^{**}	95.8	94.0^{**}	**96.2**	93.3^{**}	73.3^{**}
Scam	**99.9**	96.5^{**}	98.4^{**}	99.2^{**}	97.3^{**}	99.8	98.4^{**}	99.6	97.6^{**}	95.6^{**}
Reuters	95.9	93.8^{**}	94.0^{**}	95.5	91.9	**96.9****	95.8	96.2	96.3	90.4^{**}
20newsgroup	**93.7**	81.6^{**}	86.6^{**}	81.1^{**}	66.9^{**}	91.9	82.7^{**}	87.3^{**}	73.9^{**}	53.7^{**}
MovieReviews	75.1	66.8^{**}	74.5	28.8^{**}	57.1^{**}	**77.2**	63.0^{**}	68.9^{**}	67.5^{**}	24.8^{**}
Webmaster	**88.6**	82.5	85.6	82.5	79.1^{**}	86.7	82.0^{*}	86.8	86.7	63.8^{**}
Ads	**81.3**	73.8^{*}	72.7^{*}	70.0^{**}	19.7^{**}	78.2	71.7^{**}	72.2^{**}	63.6^{**}	17.2^{**}
Median F1	**91.1**	82.0	86.1	81.8	73.0	89.3	82.3	87.0	80.3	58.8
Avg. Rank	**1.75**	6.12	4.62	6.12	8.75	3.71	6.25	3.50	5.75	10.0
nonNLP Data.										
Sig	80.2	66.4^{**}	74.1^{*}	67.0^{*}	60.9^{**}	**80.3**	80.2	**80.3**	79.6	79.6
Reply	93.4	89.9	93.2	92.0	90.0	93.5	93.6	93.6	**94.2**	**94.2**
Adult	25.0	46.7^{**}	44.7^{**}	13.4^{**}	41.8^{**}	19.6^{**}	**49.8****	49.1^{**}	18.8^{**}	41.0^{**}
Congress	94.2	92.5^{*}	93.6	92.4	93.3^{*}	**96.0**	94.3	95.2	94.3	92.5
Credit	72.1	79.1	74.3	46.2^{**}	59.3^{**}	**79.7**	78.1	77.3	60.0^{**}	66.9
Wisc	96.8	96.4	96.3	**97.5**	96.0	97.2	96.9	96.7	97.4	95.7
Nursery	69.6	55.8^{*}	69.1	72.0	68.3	69.6	80.3^{**}	83.1^{**}	**86.3****	85.8^{**}
Median F1	80.2	79.1	74.3	72.0	68.3	80.3	80.3	83.1	**86.3**	85.8
Avg. Rank	5.57	7.00	6.42	7.42	8.28	3.71	**3.14**	**3.14**	4.28	5.71

Table 3: General Performance of Single-Pass Online Learners – F1 measures (%). PW=Positive Winnow, BW=Balanced Winnow, PA=Passive-Aggressive. The symbols * and ** indicate paired t-Test statistical significance (relative to MBW) with $p \leq 0.05$ and $p \leq 0.01$ levels, respectively.

Outline

- Part I: the basics
  - What is text classification? Why do it?
  - Representing text for classification
  - A simple, fast generative method
  - Some simple, fast discriminative methods
- Part II: advanced topics
  - Sentiment detection and subjectivity
  - Collective classification
  - Alternatives to bag-of-words

Text Classification: Examples

- Classify news stories as *World, US, Business, SciTech, Sports, Entertainment, Health, Other:* ***topical classification, few classes***
- Classify email to tech staff as *Mac, Windows, ..., Other:* ***topical classification, few classes***
- Classify email as *Spam, Other:* ***topical classification, few classes***
  - Adversary may try to defeat your categorization scheme
- Add MeSH terms to Medline abstracts
  - e.g. "Conscious Sedation" [E03.250]
  - ***topical classification, many classes***
- Classify web sites of companies by Standard Industrial Classification (SIC) code.
  - ***topical classification, many classes***
- Classify business names by industry.
- Classify student essays as *A,B,C,D,* or *F.*
- Classify pdf files as *ResearchPaper, Other*
- Classify documents as *WrittenByReagan, GhostWritten*
- Classify movie reviews as *Favorable,Unfavorable,Neutral.*
- Classify technical papers as *Interesting, Uninteresting.*
- Classify jokes as *Funny, NotFunny.*

Classifying Reviews as Favorable or Not

[Turney, ACL 2002]

- Dataset: 410 reviews from Epinions
  - Autos, Banks, Movies, Travel Destinations
- Learning method:
  - Extract 2-word phrases containing an adverb or adjective (eg "unpredictable plot")
  - Classify reviews based on average Semantic Orientation

$$\mathrm{SO}(\mathit{phrase}) = \mathrm{PMI}(\mathit{phrase}, \text{“excellent”}) - \mathrm{PMI}(\mathit{phrase}, \text{“poor”})$$

$$\mathrm{PMI}(\mathit{word}_1, \mathit{word}_2) = \log_2\left[\frac{\mathrm{p}(\mathit{word}_1 \,\&\, \mathit{word}_2)}{\mathrm{p}(\mathit{word}_1)\,\mathrm{p}(\mathit{word}_2)}\right]$$

Computed using queries to web search engine

Classifying Reviews as Favorable or Not

[Turney, ACL 2002]

Extracted Phrase	Part-of-Speech Tags	Semantic Orientation
online experience	JJ NN	2.253
low fees	JJ NNS	0.333
local branch	JJ NN	0.421
small part	JJ NN	0.053
online service	JJ NN	2.780
printable version	JJ NN	-0.705
direct deposit	JJ NN	1.288
well other	RB JJ	0.237
inconveniently located	RB VBN	-1.541
other bank	JJ NN	-0.850
true service	JJ NN	-0.732
Average Semantic Orientation		0.322

Classifying Reviews as Favorable or Not

[Turney, ACL 2002]

Table 5. The accuracy of the classification and the correlation of the semantic orientation with the star rating.

Domain of Review	Accuracy	Correlation
Automobiles	84.00 %	0.4618
Honda Accord	83.78 %	0.2721
Volkswagen Jetta	84.21 %	0.6299
Banks	80.00 %	0.6167
Bank of America	78.33 %	0.6423
Washington Mutual	81.67 %	0.5896
Movies	65.83 %	0.3608
The Matrix	66.67 %	0.3811
Pearl Harbor	65.00 %	0.2907
Travel Destinations	70.53 %	0.4155
Cancun	64.41 %	0.4194
Puerto Vallarta	80.56 %	0.1447
All	74.39 %	0.5174

Guess majority class always: 59% accurate.

Classifying Movie Reviews

[Pang et al, EMNLP 2002]

	Features	# of features	frequency or presence?	NB	ME	SVM
(1)	unigrams	16165	freq.	**78.7**	N/A	72.8
(2)	unigrams	"	pres.	81.0	80.4	**82.9**
(3)	unigrams+bigrams	32330	pres.	80.6	80.8	**82.7**
(4)	bigrams	16165	pres.	77.3	**77.4**	77.1
(5)	unigrams+POS	16695	pres.	81.5	80.4	**81.9**
(6)	adjectives	2633	pres.	77.0	**77.7**	75.1
(7)	top 2633 unigrams	2633	pres.	80.3	81.0	**81.4**
(8)	unigrams+position	22430	pres.	81.0	80.1	**81.6**

700 movie reviews (ie all in same domain); Naïve Bayes, MaxEnt, and linear SVMs; accuracy with different representations x for a document

Interestingly, the off-the-shelf methods work well…perhaps better than Turney's method.

Classifying Movie Reviews

[Pang et al, EMNLP 2002]

MaxEnt classification:

• Assume the classifier is same form as Naïve Bayes, which can be written:

$$\Pr(y \mid w_1, w_2, \ldots., w_N) = \frac{1}{Z} \prod_i \lambda_i f(y, w_i)$$

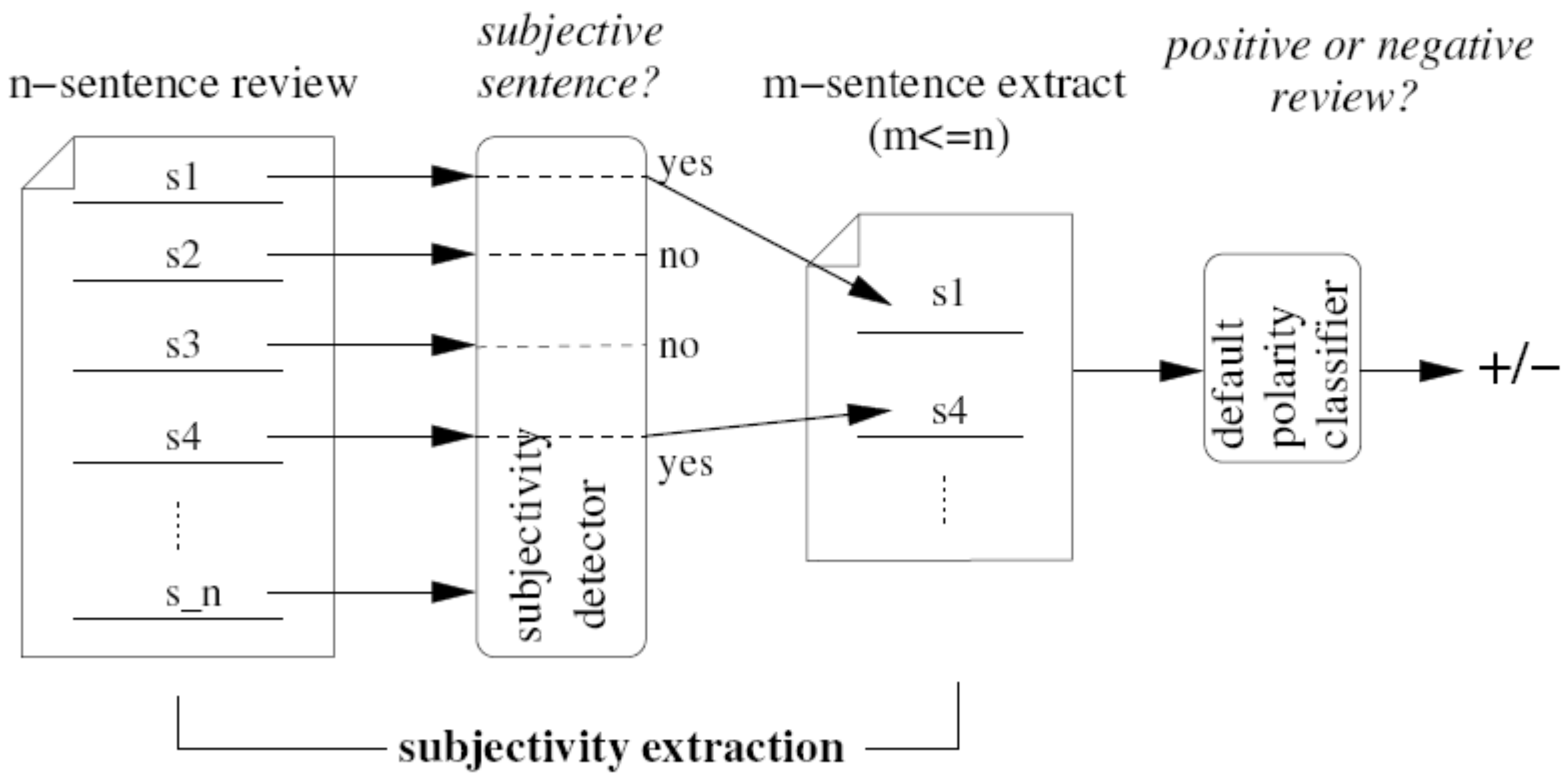
•Set weights (**λ**'s) to maximize probability of the training data:

$$\prod_{(x_j, y_j) \in D} \Pr(y_j \mid x_j) + \underbrace{\Pr(\lambda \mid Q)}_{\textbf{prior on parameters}}$$

Classifying Movie Reviews

[Pang et al, ACL 2004]

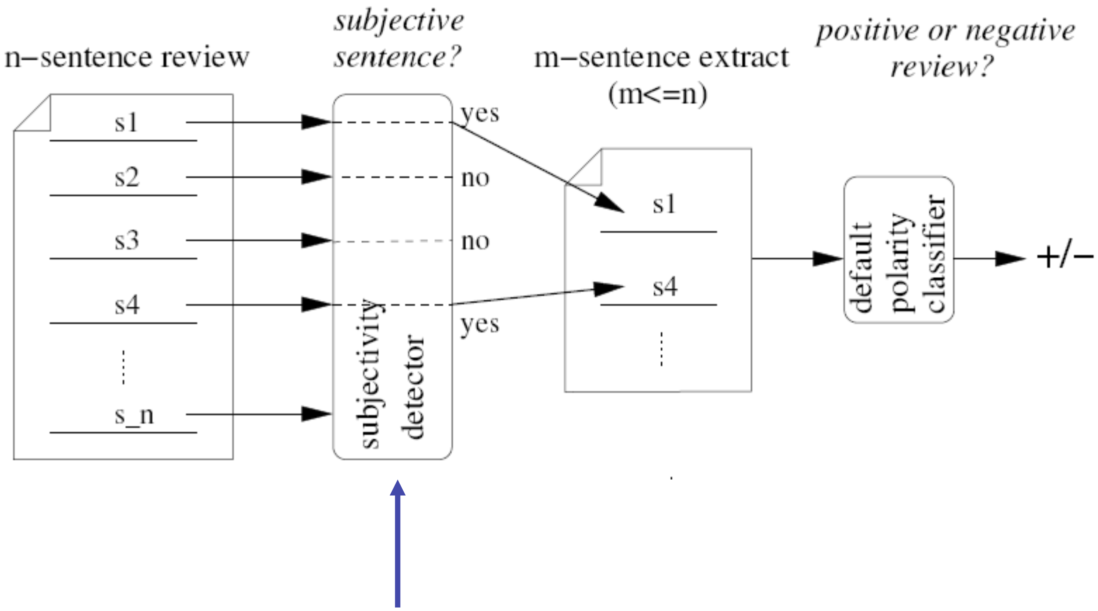
Idea: like Turney, focus on "polar" sections: *subjective sentences*



Classifying Movie Reviews

[Pang et al, ACL 2004]

Idea: like Turney, focus on "polar" sections: *subjective sentences*



Dataset for subjectivity: Rotten Tomatoes (+), IMDB plot reviews (-)
Apply ML to build a sentence classifier
Try and force *nearby sentences* to have similar subjectivity

"Fearless" allegedly marks Li's last turn as a martial arts movie star--at 42, the ex-wushu champion-turned-actor is seeking a less strenuous on-camera life--and it's based on the life story of one of China's historical sports heroes, Huo Yuanjia. Huo, a genuine legend, lived from 1868-1910, and his exploits as a master of wushu (the general Chinese term for martial arts) raised national morale during the period when beleaguered China was derided as "The Sick Man of the East."

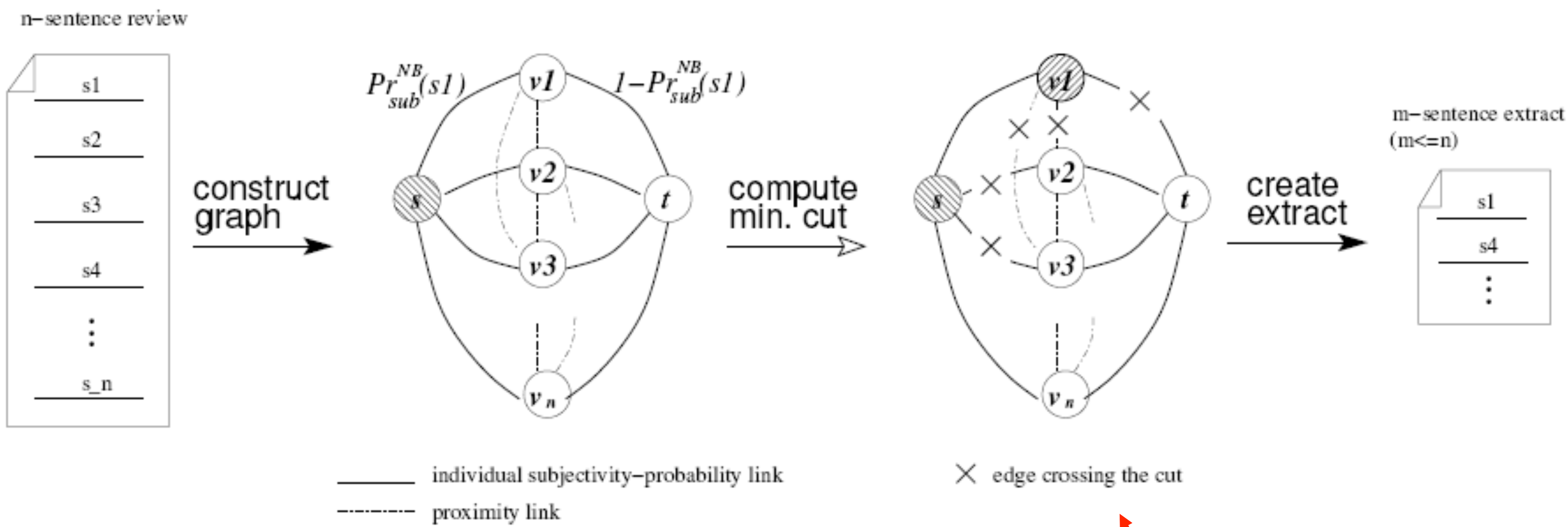
"Fearless" shows Huo's life story in highly fictionalized terms, though the movie's most dramatic sequence--at the final Shanghai tournament, where Huo takes on four international champs, one by one--is based on fact. It's a real old-fashioned movie epic, done in director Ronny Yu's ("The Bride with White Hair") usual flashy, Hong Kong-and-Hollywood style, laced with spectacular no-wires fights choreographed by that Bob Fosse of kung fu moves, Yuen Wo Ping ("Crouching Tiger" and "The Matrix"). Dramatically, it's on a simplistic level. But you can forgive any historical transgressions as long as the movie keeps roaring right along.

"Fearless" allegedly marks Li's last turn as a martial arts movie star--at 42, the ex-wushu champion-turned-actor is seeking a less strenuous on-camera life--and it's based on the life story of one of China's historical sports heroes, Huo Yuanjia. Huo, a genuine legend, lived from 1868-1910, and his exploits as a master of wushu (the general Chinese term for martial arts) raised national morale during the period when beleaguered China was derided as "The Sick Man of the East."

"Fearless" shows Huo's life story in highly fictionalized terms, though the movie's most dramatic sequence--at the final Shanghai tournament, where Huo takes on four international champs, one by one--is based on fact. It's a real old-fashioned movie epic, done in director Ronny Yu's ("The Bride with White Hair") usual flashy, Hong Kong-and-Hollywood style, laced with spectacular no-wires fights <u>choreographed by that Bob Fosse of kung fu moves, Yuen Wo Ping ("Crouching Tiger" and "The Matrix").</u> Dramatically, it's on a simplistic level. But you can forgive any historical transgressions as long as the movie keeps roaring right along.

Classifying Movie Reviews

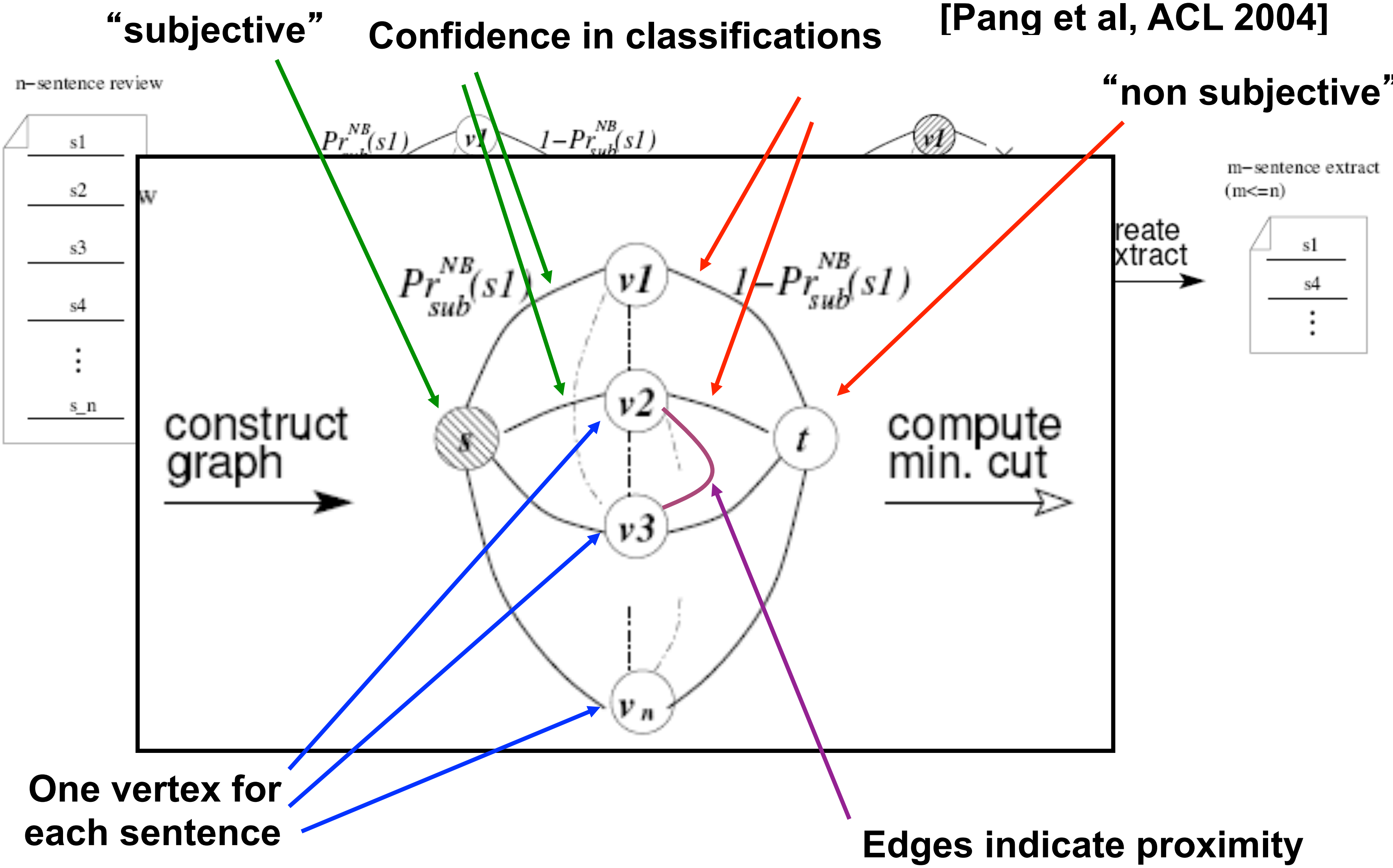
[Pang et al, ACL 2004]



Dataset: Rotten Tomatoes (+), IMDB plot reviews (-)
Apply ML to build a sentence classifier
Try and force *nearby sentences* to have similar subjectivity: use methods to find <u>minimum cut</u> on a constructed graph

Classifying Movie Reviews

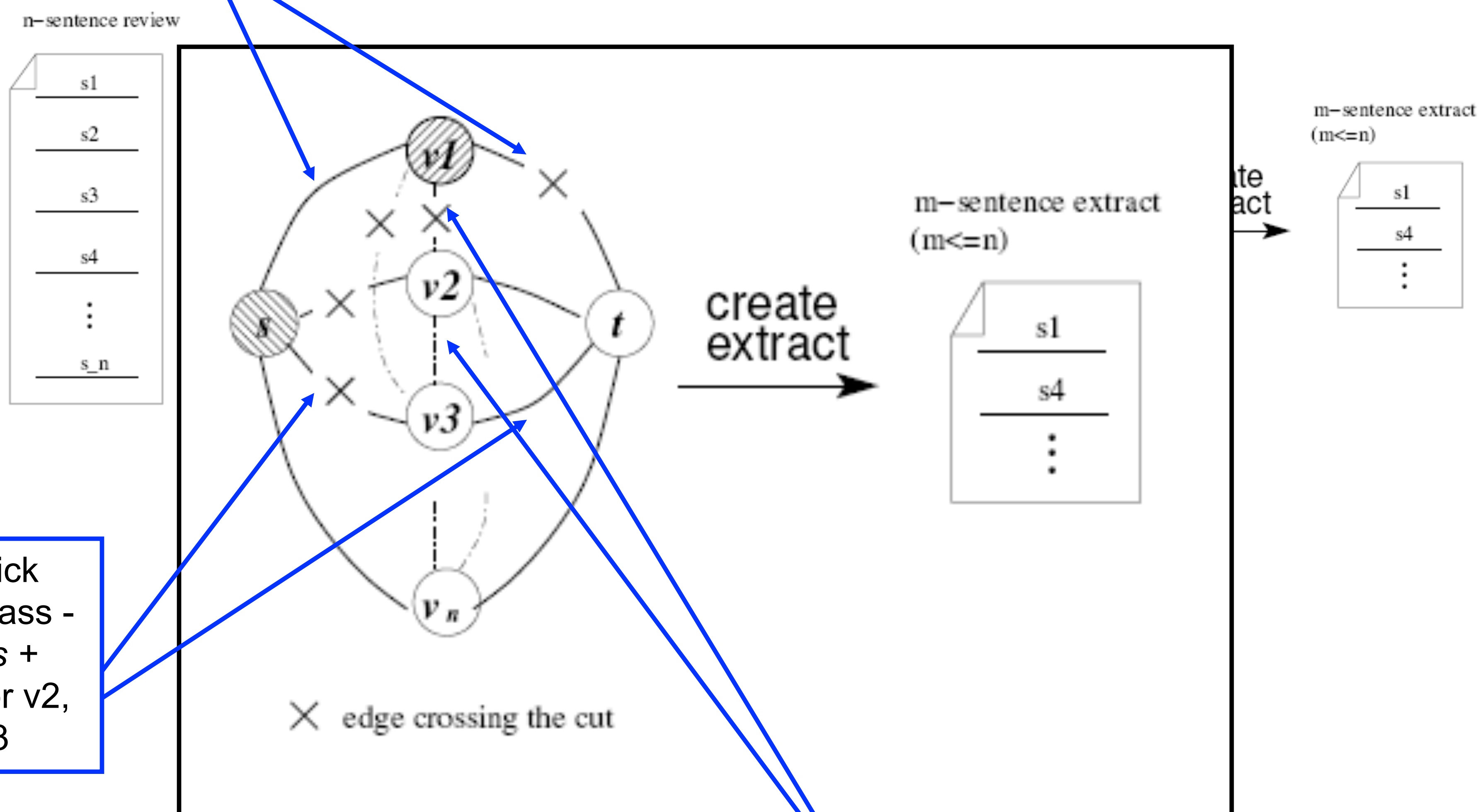
[Pang et al, ACL 2004]



Classifying Movie Reviews

[Pang et al, ACL 2004]

Pick class + vs – for v1

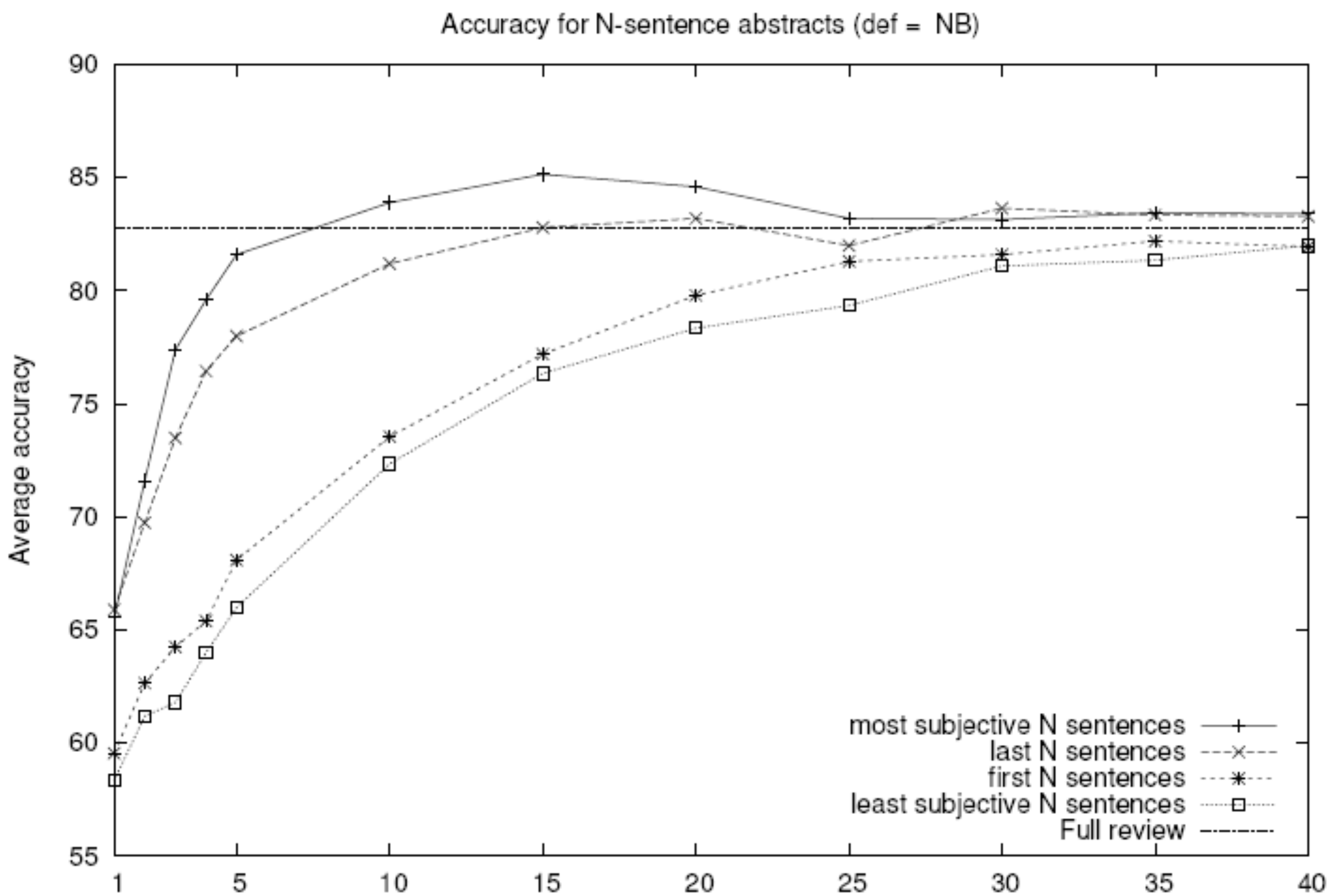


Pick class - *vs* + for v2, v3

Retained f(v2)=f(v3), but not f(v2)=f(v1)

Classifying Movie Reviews

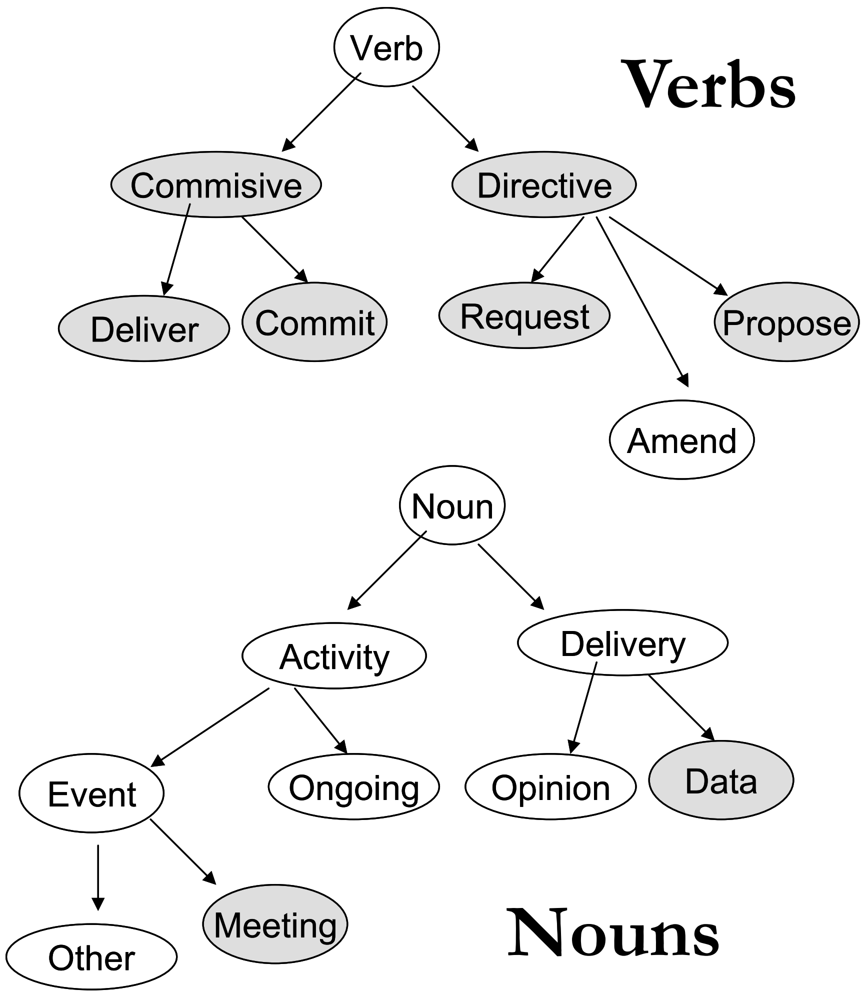
[Pang et al, ACL 2004]



Outline

- Part I: the basics
  - What is text classification? Why do it?
  - Representing text for classification
  - A simple, fast generative method
  - Some simple, fast discriminative methods
- Part II: advanced topics
  - Sentiment detection and subjectivity
  - Collective classification
  - Alternatives to bag-of-words

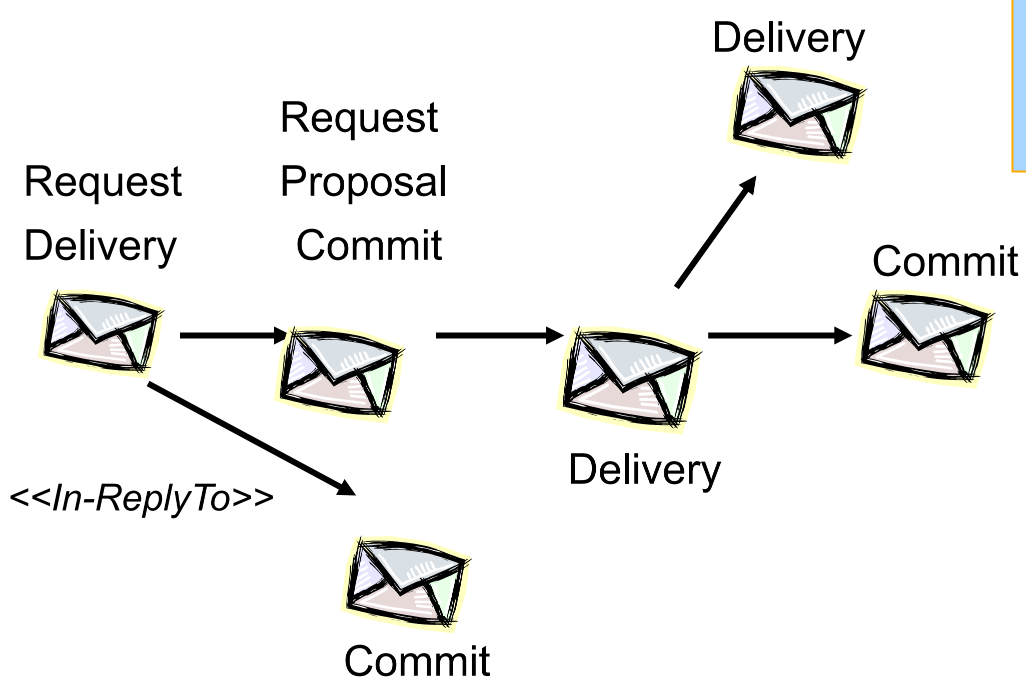
Classifying Email into Acts



- From EMNLP-04, *Learning to Classify Email into Speech Acts*, Cohen-Carvalho-Mitchell
- An Act is described as a verb-noun pair (e.g., propose meeting, request information) - Not all pairs make sense. One single email message may contain multiple acts.
- Try to describe commonly observed behaviors, rather than all possible speech acts in English. Also include non-linguistic usage of email (e.g. delivery of files)

Idea: Predicting Acts from Surrounding Acts

Example of Email Sequence

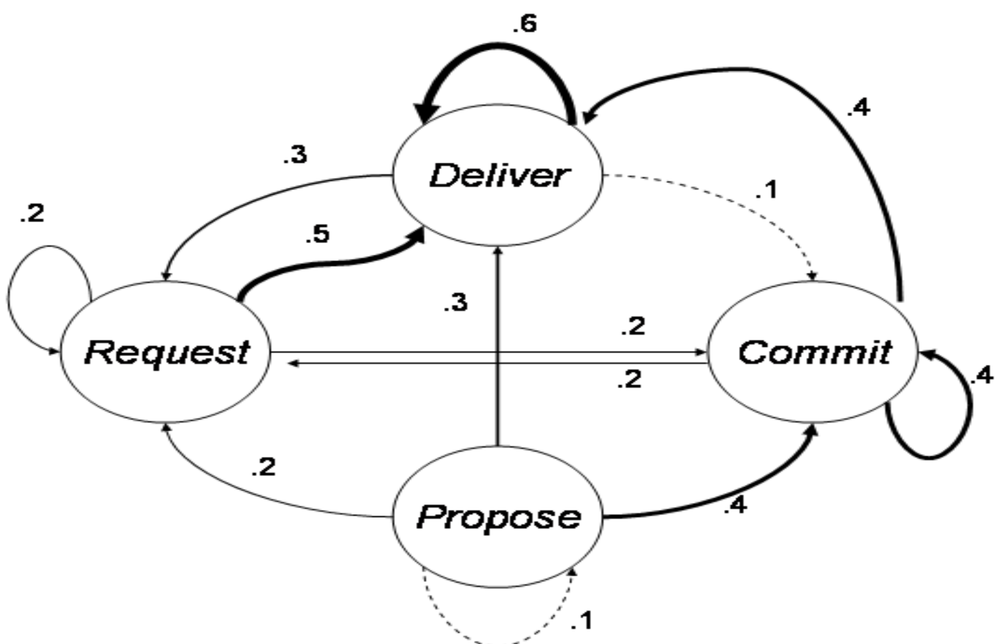


- Lots of *information about the acts in a message* by looking at the acts in the *parent & child messages*.

- Acts in parent/child messages do **not** tend to be the *same* as acts in message
  - So, mincut is not appropriate technique.

Evidence of Sequential Correlation of Acts

- Transition diagram for most common verbs from CSPACE corpus (Kraut & Fussell)
- Act sequence patterns: (Request, Deliver+), (Propose, Commit+, Deliver+), (Propose, Deliver+), most common act was Deliver

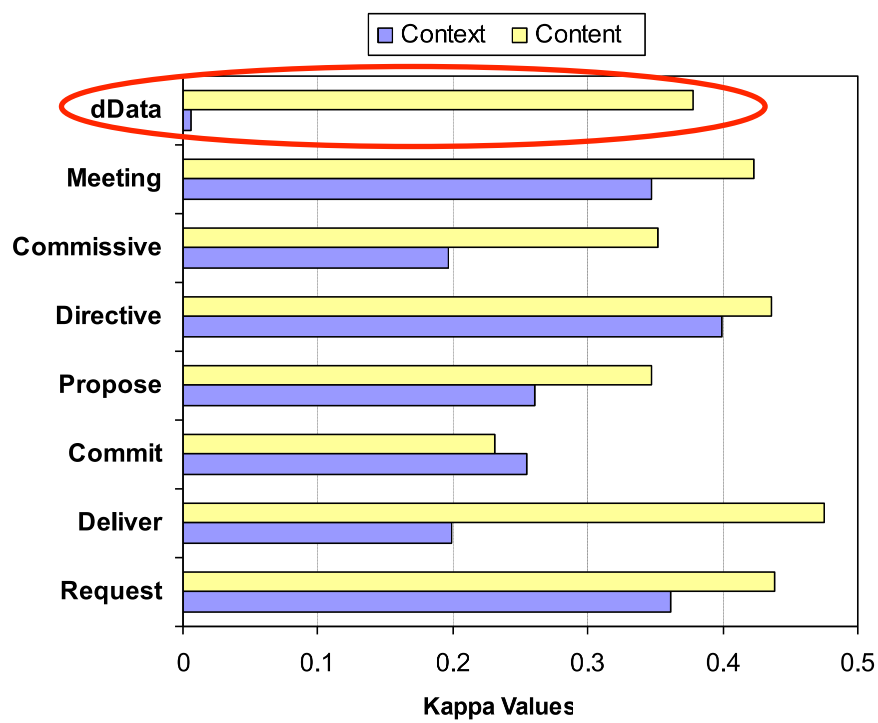


Data: CSPACE Corpus

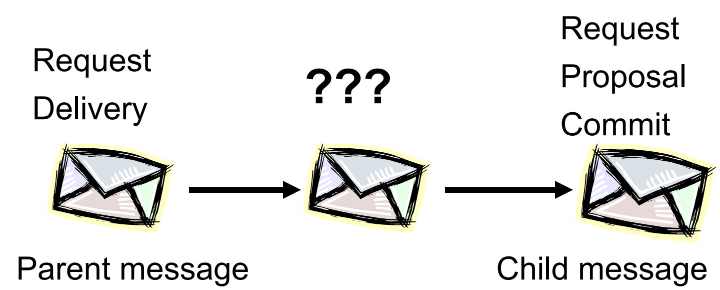
- Few large, free, natural email corpora are available
- CSPACE corpus (Kraut & Fussell)
  - Emails associated with a semester-long project for Carnegie Mellon MBA students in 1997
  - 15,000 messages from 277 students, divided in 50 teams (4 to 6 students/team)
  - Rich in task negotiation.
  - More than 1500 messages (from 4 teams) were labeled in terms of “Speech Act”.
  - One of the teams was double labeled, and the inter-annotator agreement ranges from 72 to 83% (Kappa) for the most frequent acts.

Content versus Context

- **Content:** Bag of Words features only
- **Context:** *Parent and Child Features* only (table below)
- 8 MaxEnt classifiers, trained on 3F2 and tested on 1F3 team dataset
- Only 1st child message was considered (vast majority – more than 95%)



Kappa Values on 1F3 using Relational (Context) features and Textual (Content) features.



Parent Boolean Features		*Child Boolean Features*
Parent_Request, Parent_Deliver, Parent_Commit, Parent_Propose, Parent_Directive, Parent_Commissive Parent_Meeting, Parent_dData		Child_Request, Child_Deliver, Child_Commit, Child_Propose, Child_Directive, Child_Commissive, Child_Meeting, Child_dData

Set of Context Features (Relational)

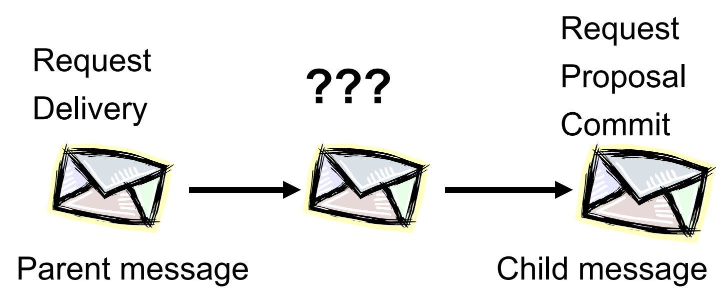
Content versus Context

- **Content:** Bag of Words features only
- **Context:** *Parent and Child Features* only (table below)
- 8 MaxEnt classifiers, trained on 3F2 and tested on 1F3 team dataset
- Only 1st child message was considered (vast majority – more than 95%)

Ok, that's a nice experiment: but how can we **use** the parent/child features?

• To classify **x** we need to classify parent(**x**) and firstChild(**x**)

• To classify firstChild(**x**) we need to classify parent(firstChild(**x**))=**x**



Parent Boolean Features		*Child Boolean Features*
Parent_Request, Parent_Deliver, Parent_Commit, Parent_Propose, Parent_Directive, Parent_Commissive Parent_Meeting, Parent_dData		Child_Request, Child_Deliver, Child_Commit, Child_Propose, Child_Directive, Child_Commissive, Child_Meeting, Child_dData

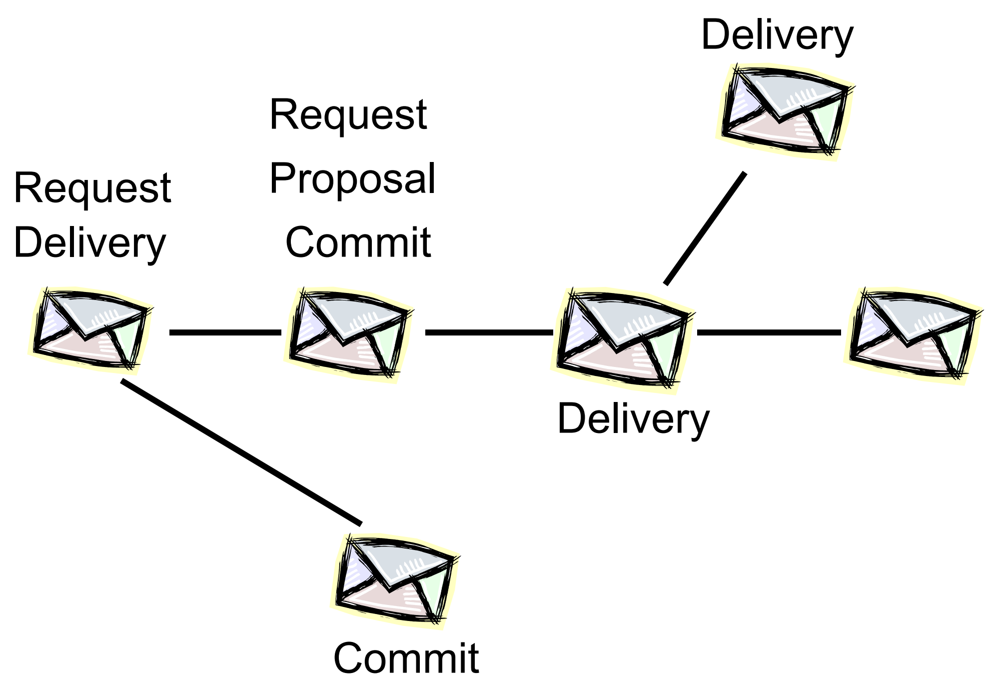
Set of Context Features (Relational)

Collective Classification using Dependency Networks

Dependency networks are probabilistic graphical models in which the full joint distribution of the network is approximated with a set of *conditional distributions* that can be learned independently. The conditional probability distributions in a DN are calculated for each node given its neighboring nodes (its *Markov blanket*).

$$\Pr(\vec{X}) = \prod_i \Pr(X_i \mid NeighborSet(X_i))$$

- No acyclicity constraint. Simple parameter estimation – approximate inference (Gibbs sampling)
- Closely related to pseudo-likelihood
- In this case, NeighborSet(x) = Markov blanket = parent message and child message



Collective Classification algorithm
(based on Dependency Networks Model)

Learn

1- For each of the 8 email-acts, build a *local classifier* LC_{act} from the training set

Classify

2- Initialize the test set with email-act classes based on a content-only classifier.

3- For each iteration j=0 to T:

  3.1- Update *Confidence Threshold* (%) $\theta = 100 - j$;

  3.2- If ($\theta < 50$), make $\theta = 50$;

  3.3- For every email msg in test set, in chronological order:

    3.3.1- For each email-act class:

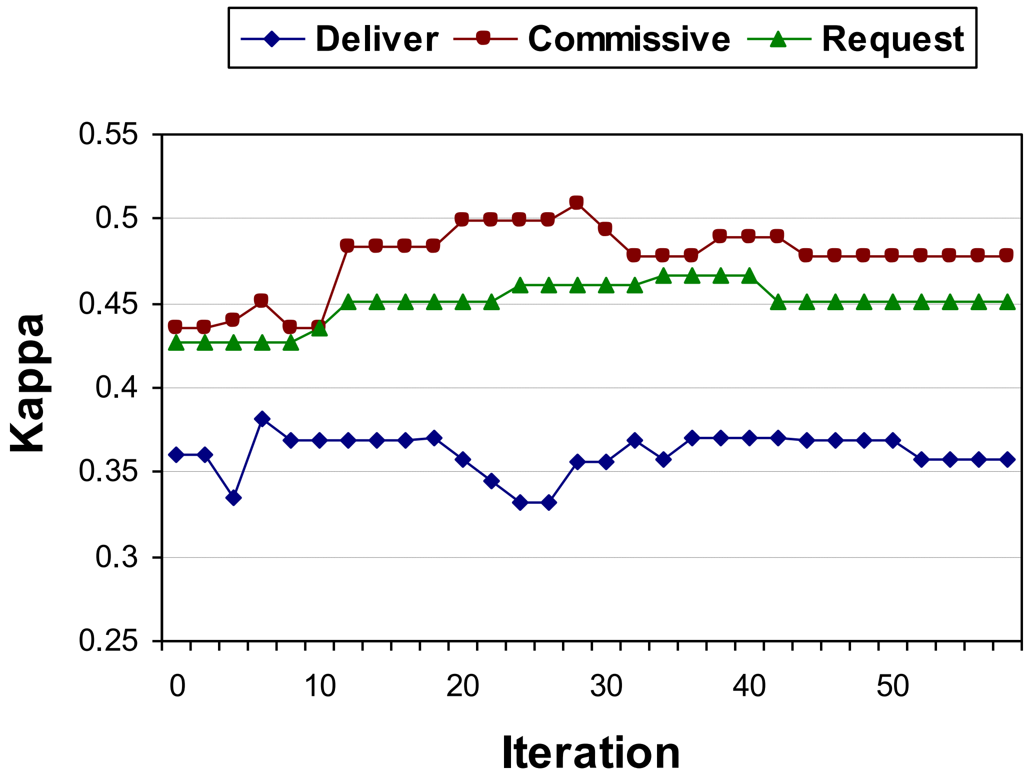
      3.3.1.1- obtain confidence(act, msg) from LC_{act}(msg)

      3.3.1.2- if (confidence(act,msg) > θ), update email-act of msg

  3.4- Calculate performance on this iteration

4- Output final inferences and calculate final performance

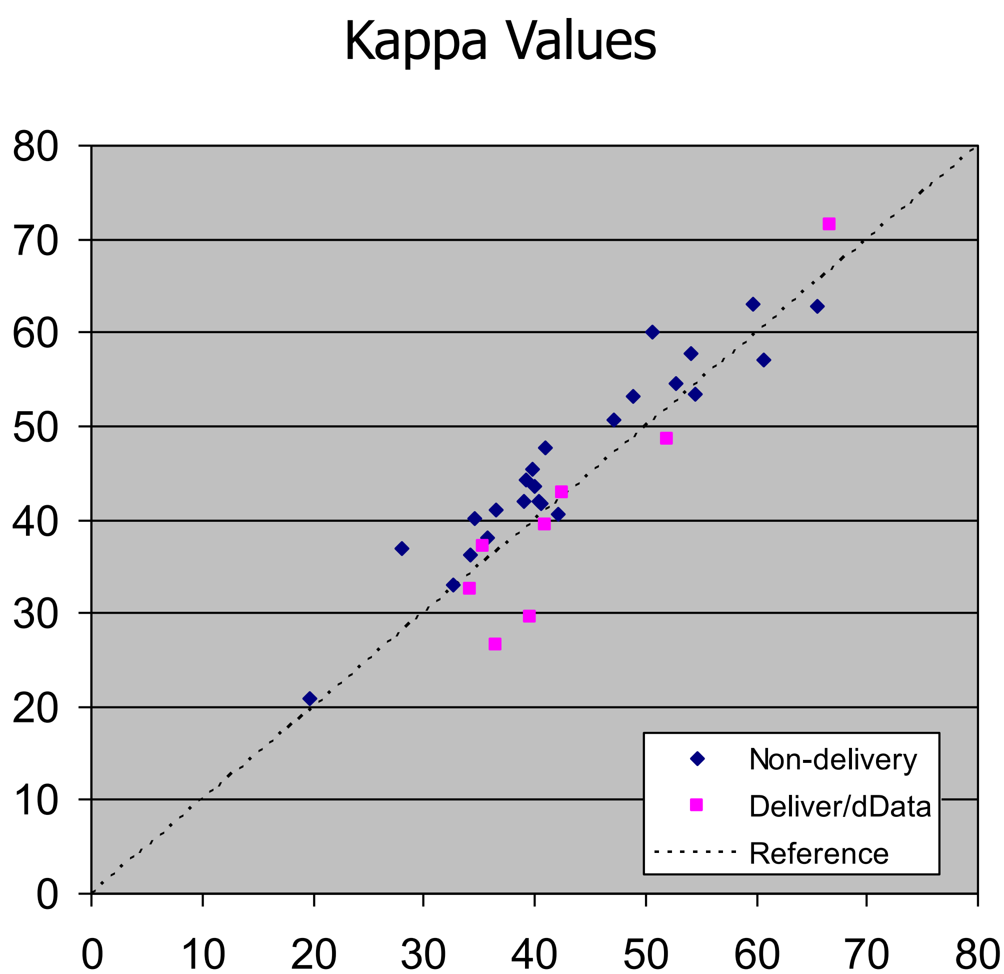
Agreement versus Iteration



- **Kappa versus iteration on 1F3 team dataset, using classifiers trained on 3F2 team data.**

Leave-one-team-out Experiments

- Deliver and dData performance usually decreases
- Associated with data distribution, FYI, file sharing, etc.
- For "non-delivery", improvement in avg. Kappa is statistically significant (p=0.01 on a two-tailed T-test)



Outline

- Part I: the basics
  - What is text classification? Why do it?
  - Representing text for classification
  - A simple, fast generative method
  - Some simple, fast discriminative methods
- Part II: advanced topics
  - Sentiment detection and subjectivity
  - Collective classification
  - Alternatives to bag-of-words

Text Representation for Email Acts

[Carvalho & Cohen, TextActs WS 2006]

Document → Preprocess → Word n-grams → Feature Selection

Symbol	Pattern
[number]	any sequence of numbers
[hour]	[number]:[number]
[wwhh]	"why, where, who, what, or when"
[day]	the strings "Monday, Tuesday, ..., or Sunday"
[day]	the strings "Mon, Tue, Wed, ..., or Sun"
[pm]	the strings "P.M., PM, A.M. or AM"
[me]	the pronouns "me, her, him, us or them"
[person]	the pronouns "I, we, you, he, she or they"
[aaafter]	the strings "after, before or during"
[filetype]	the strings ".doc, .pdf, .ppt, .txt, or .xls"

Table 1: Some PreProcessing Substitution Patterns

1-gram
?
please
[wwhh]
could
do
can
of
[me]

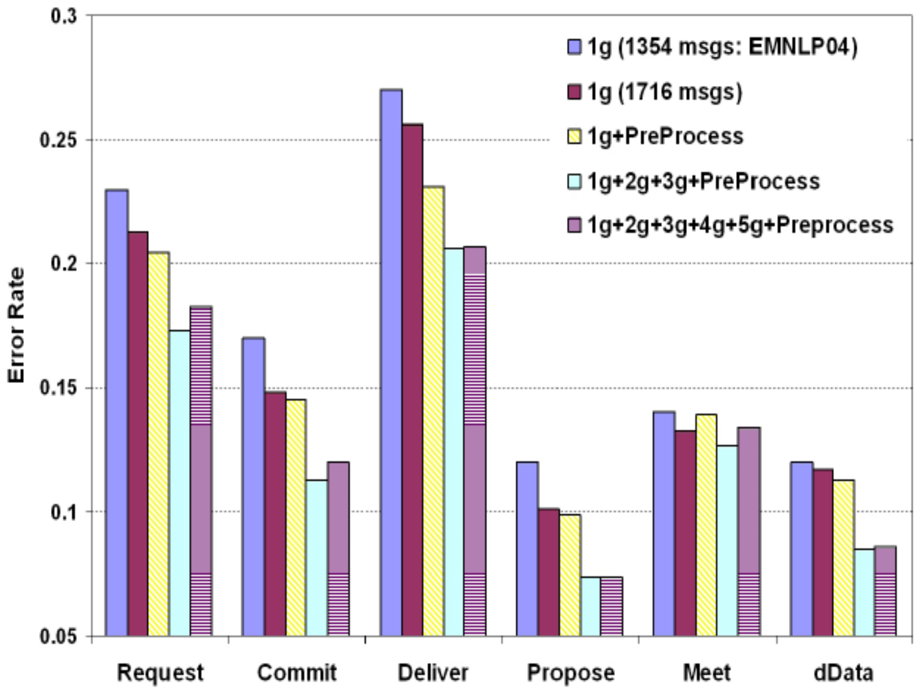
3-gram
[person] need to
[wwhh] do [person]
let [me] know
would [person]
do [person] think
are [person] meeting
could [person] please
do [person] need

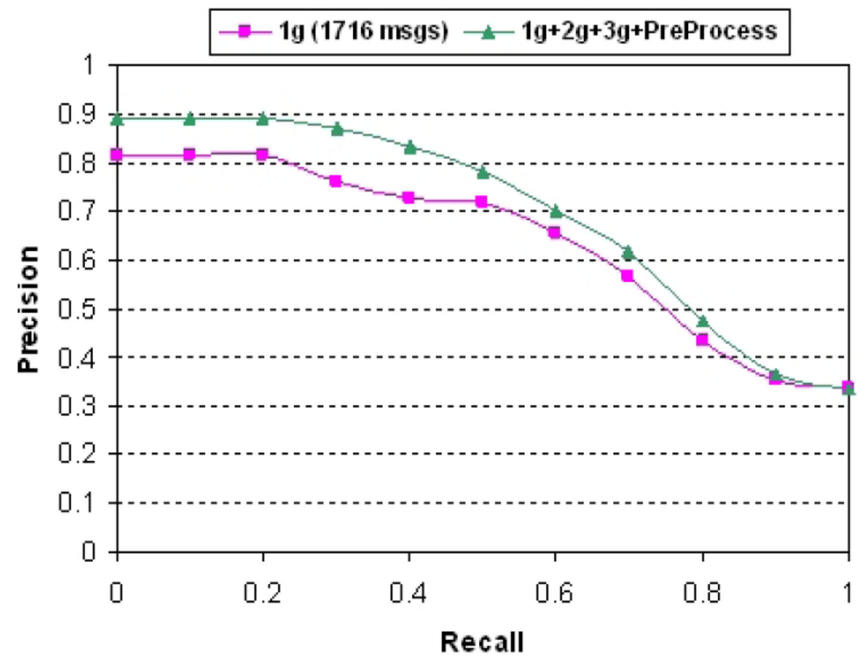
5-gram
[wwhh] do [person] think ?
let [me] know [wwhh] [person]
a call [number]-[number]
give [me] a call [number]
please give give [me] a call
[person] would be able to
take a look at it
[person] think [person] need to

Request	Commit	Meeting
[wwhh] do [person] think	is good for [me]	[day] at [hour] [pm]
do [person] need to	is fine with [me]	on [day] at [hour]
and let [me] know	i will see [person]	[person] can meet at
call [number]-[number]	i think i can	[person] meet at [hour]
would be able to	i will put the	will be in the
[person] think [person] need	i will try to	is good for [me]
let [me] know [wwhh]	i will be there	to meet at [hour]
do [person] think ?	will look for [person]	at [hour] in the
[person] need to get	$[number] per person	[person] will see [person]
? [person] need to	am done with the	meet at [hour] in
a copy of our	at [hour] i will	[number] at [hour] [pm]
do [person] have any	[day] is fine with	to go over the
[person] get a chance	each of us will	[person] will be in
[me] know [wwhh]	i will bring copies	let's plan to meet
that would be great	i will do the	meet at [hour] [pm]

dData	Propose	Deliver
– forwarded message begins	[person] would like to	forwarded message begins here
forwarded message begins here	would like to meet	[number] [number] [number] [number]
is in my public	please let [me] know	is good for [me]
in my public directory	to meet with [person]	if [person] have any
[person] have placed the	[person] meet at [hour]	if fine with me
please take a look	would [person] like to	in my public directory
[day] [hour] [number] [number]	[person] can meet tomorrow	[person] will try to
[number] [day] [number] [hour]	an hour or so	is in my public
[date] [day] [number] [day]	meet at [hour] in	will be able to
in our game directory	like to get together	just wanted to let
in the etc directory	[hour] [pm] in the	[pm] in the lobby
the file name is	[after] [hour] or [after]	[person] will be able
is in our game	[person] will be available	please take a look
fyi – forwarded message	think [person] can meet	can meet in the
just put the file	was hoping [person] could	[day] at [hour] is
my public directory under	do [person] want to	in the commons at

Results





Compare to Pang et al for movie reviews. Do n-grams help or not?

Outline

- Part I: the basics
  - What is text classification? Why do it?
  - Representing text for classification
  - A simple, fast generative method
  - Some simple, fast discriminative methods
- Part II: advanced topics
  - Sentiment detection and subjectivity
  - Collective classification
  - Alternatives to bag-of-words
- Part III: summary/conclusions

Summary & Conclusions

- There are many, many applications of text classification
- Topical classification is fairly well understood
  - Most of the information is in individual words
  - Very fast and simple methods work well
- In many applications, *classes are not topics*
  - Sentiment detection/polarity
  - Subjectivity/opinion detection
  - Detection of user intent (e.g., speech acts)
- In many applications, *distinct classification decisions are interdependent*
  - Reviews: Subjectivity of nearby sentences
  - Email: Intent of parent/child messages in a thread
  - Web: Topics of web pages linked to/from a page
  - Biomedical text: Topics of papers that cite/are cited by a paper

- Lots of prior work to build on, lots of prior experimentation to consider
- Don't be afraid of topic classification problems
  - Reliably labeled data can be hard to find in some domains
- For non-topic TC, you may need to explore different *document representations* and/or different *learning methods.*
  - We don't know the answers here
- Consider "collective classification" methods when there are strong dependencies.