Lecture 3: Spam Filtering

Siddharth Garg

sg175@nyu.edu

Spam Detection: Features

- Recall features used in the UCI Spam database

  48 continuous real [0,100] attributes of type word_freq_WORD

- Even easier way to encode features:
  - $x_i = 1$ if term *i* appears in a document; 0 otherwise
  - Boolean features

- Assume M Boolean features, x = (x_1, x_2,…, x_M)
  - We want to map this M-dimensional Boolean input to a Boolean output *y*
  - *Thoughts?*
  - Instead of using LR or SVM we will start with an even simpler approach referred to as "Naiive Bayes"

Ref: Metsis, Vangelis, Ion Androutsopoulos, and Georgios Paliouras. "Spam filtering with naive bayes-which naive bayes?." In *CEAS*, vol. 17, pp. 28-69. 2006.

Naiive Bayes for Spam Filtering

- Assume M Boolean feature, x = ($x_1, x_2, \ldots, x_M$)
- Each email is either {s=spam,l=legit}
- We begin by computing:

"Bernoulli Naiive Bayes"

$$P\{spam \mid x\} = \frac{P\{x \mid spam\} * P\{spam\}}{P\{x\}}$$

Bayes Rule $$\boxed{P\{A \cap B\} = P\{A \mid B\} * P\{B\}}$$

Ref: Metsis, Vangelis, Ion Androutsopoulos, and Georgios Paliouras. "Spam filtering with naive bayes-which naive bayes?." In *CEAS*, vol. 17, pp. 28-69. 2006.

Naiive Bayes for Spam Filtering

- We begin by computing:

$$P\{spam \mid x\} = \frac{P\{x \mid spam\} * P\{spam\}}{P\{x\}}$$

$$P\{x_1, x_2, .., x_M \mid spam\} = P\{x_1 \mid spam\} * P\{x_2 \mid spam\} * .. * P\{x_M \mid spam\}$$

Assuming that term occurrences are independent (given class)!

Is this a reasonable assumption?

Naiive Bayes for Spam Filtering

$$P\{x_1 \mid spam\} * P\{x_2 \mid spam\} * .. * P\{x_M \mid spam\}$$

How do we estimate this from the training dataset?

$$P\{x_1 = 1 \mid spam\} = p_{i,s}$$

=(#Spam emails that contain term 1)/(#spam emails)

What happens if term 1 never occurred in any spam email in the training dataset?

Laplacian Smoothing

$p_{i,s}$ = ~~(#Spam emails that contain term 1)/(#spam emails)~~

= (#Spam emails that contain term 1+1)/(#spam emails+2)

Equivalent to assuming two addition spam emails in the training dataset, of which on contains all terms and the other is empty

$$P\{x_1 = 0 \mid spam\} = 1 - p_{i,s}$$

$$\boxed{P\{x_1, x_2, .., x_M \mid spam\} = \prod_{i=1}^{M} p_{i,s}^{x_i} \; (1 - p_{i,s})^{1-x_i}}$$

Naiive Bayes for Spam Filtering

$$P\{x\} = P\{spam\} * P\{x \mid spam\} + P\{legit\} * P\{x \mid legit\}$$

$$P\{spam \mid x\} = \frac{P\{x \mid spam\} * P\{spam\}}{P\{x\}}$$ Vs. $$P\{legit \mid x\} = \frac{P\{x \mid legit\} * P\{legit\}}{P\{x\}}$$

Or:

$$P\{spam \mid x\} \geq threshold$$

In-Class Exercise

\# Spam Emails in Training Dataset: 50

\# Legit Emails in Training Dataset: 100

Word/Term	#Spam Emails with Term	#Legit Emails with Term
"FREE"	40	0
"George"	0	20
"and"	40	80

Test Email: $\{x_{FREE}, x_{GEORGE}, x_{and}\} = \{1,1,0\}$

$$P\{spam \mid x\} = \frac{P\{x \mid spam\} * P\{spam\}}{P\{x\}} \quad \text{Vs.} \quad P\{legit \mid x\} = \frac{P\{x \mid legit\} * P\{legit\}}{P\{x\}}$$

Solution

Spam Emails in Training Dataset: 50

Legit Emails in Training Dataset: 100

Word/Term	#Spam Emails with Term	#Legit Emails with Term
"FREE"	40	0
"George"	0	20
"and"	40	80

Test Email: $\{x_{FREE}, x_{GEORGE}, x_{and}\} = \{1,1,0\}$

$$P\{spam \mid x\} =$$

Vs.

$$P\{legit \mid x\} =$$

Spam Detection: Occurences

- Recall features used in the UCI Spam database

  48 continuous real [0,100] attributes of type word_freq_WORD

- Let's consider a different representation that is closer to the UCI spambase features: Term Frequencies (TF)
  - x_i # times term *i* appears in a document ($x_i \in \aleph$)
  - Each document is represented by x = (x_1, x_2, …, x_M), a vector of term frequencies
  - We will again use a Naïve Bayes approach to classify documents as either spam or legit
    - "**Multinomial Naïve Bayes**"

Applying Bayes Rule

$$P\{spam \mid x\} = \frac{P\{x \mid spam\} * P\{spam\}}{P\{x\}}$$

$$P\{x_1 \mid spam\} * P\{x_2 \mid spam\} * .. * P\{x_M \mid spam\}$$

Independence assumption shows up again!

But how do we estimate the probability : $P\{x_1 = t \mid spam\}$

What if there is no document in the training dataset where term 1 occurs *t* times?

"Bag of Words" Model

Bag containing *M* terms

Term *i* picked with probability $p_{i,s}$

$$\sum_{i=1}^{M} p_{i,s} = 1$$

Term i picked with probability $p_{i,s}$

Term i picked with probability $p_{i,s}$

1

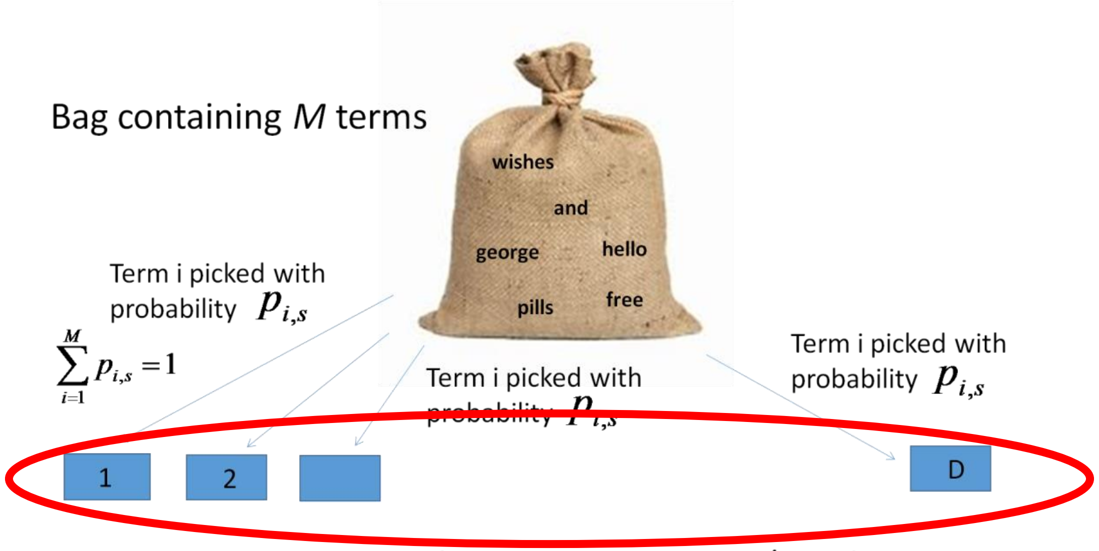
2

D

Spam Email Containing D terms/words

Likelihood Estimation

- Say you have a spam e-mail of length D generated using the BoW model



Spam Email Containing D terms/words

Feature vector x

$$P\{x \mid spam, D\} = \prod_{i=1}^{M} (p_{i,s})^{x_i}$$

Assuming term and positional independence

Are we done?

Likelihood Estimation

- Recall that the BoW model does not keep track of the positions in which terms appear
  - We must account for all possible ways of arranging
    - x_1 instances of term 1 and
    - x_2 instances of term 2 and
    - … x_M instances of term M into D locations

$$ {}^{D}C_{x_1} * {}^{D-x_1}C_{x_2} * .. * {}^{D-x_1-x_2-..x_{M-1}}C_{x_M} $$

Choose x_1 locations from a total of D locations

Choose x_2 locations from remaining D- x_1 locations

Likelihood Estimation

$$ {}^{D}C_{x_1} * {}^{D-x_1}C_{x_2} * .. * {}^{D-x_1-x_2-..x_{M-1}}C_{x_M} = \frac{D!}{x_1!(D\cancel{-}x_1)!} * \frac{(D\cancel{-}x_1)!}{x_2!(D\cancel{-}x_1-x_2)!} ...1 $$

$$ = \frac{D!}{x_1!x_2!..x_M!} $$

$$ P\{x \mid spam, D\} = D!\prod_{i=1}^{M}\frac{(p_{i,s})^{x_i}}{x_1!} $$

Typo: this should be x_{i}

Note that this expression is conditioned on the length of the e-mail *D*. In practice, emails can be of varying lengths.

Accounting for Document Length

$$P\{x \mid spam\} = P\{x \mid spam, D\}P\{D \mid spam\} = \boxed{P\{x \mid spam, D\}P\{D\}}$$

Assume email length is independent of whether email is spam or legit.

Putting it all together:

$$P\{spam \mid x\} = \frac{P\{x \mid spam, D\}P\{\cancel{D}\}P\{spam\}}{P\{\cancel{x}\}}$$ **Vs.** $$P\{legit \mid x\} = \frac{P\{x \mid legit, D\}P\{\cancel{D}\}P\{legit\}}{P\{\cancel{x}\}}$$

Estimating Model Parameters

Note: this is **not** the probability of term *i* appearing in a spam document

Bag containing *M* terms

Term i picked with probability $p_{i,s}$

$$\sum_{i=1}^{M} p_{i,s} = 1$$

Term i picked with probability $p_{i,s}$

Term i picked with probability $p_{i,s}$

Probability that a randomly selected word in a spam email is term *i*

1 2 D

Spam Email Containing D terms/words

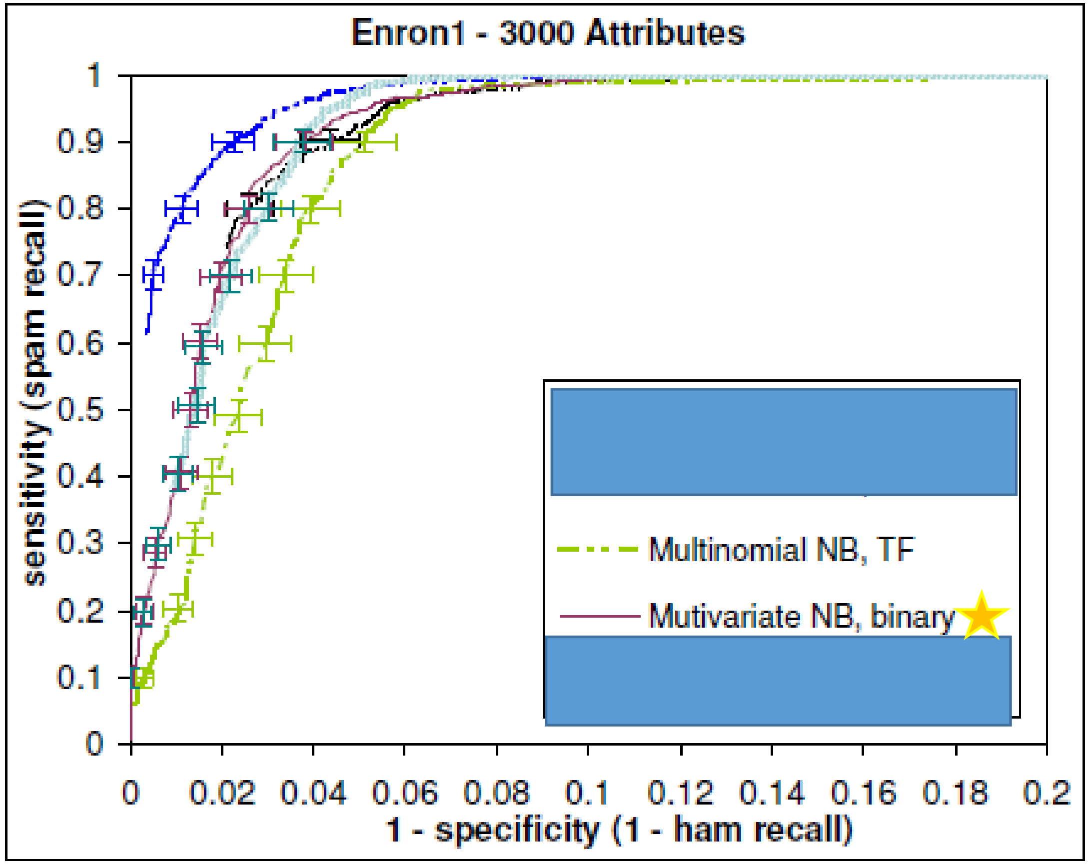
$$p_{i,s} = \frac{\#\text{ occurrences of term } i \text{ in spam}}{\sum_{i=1}^{M} \#\text{ occurrences of term } i \text{ in spam}}$$

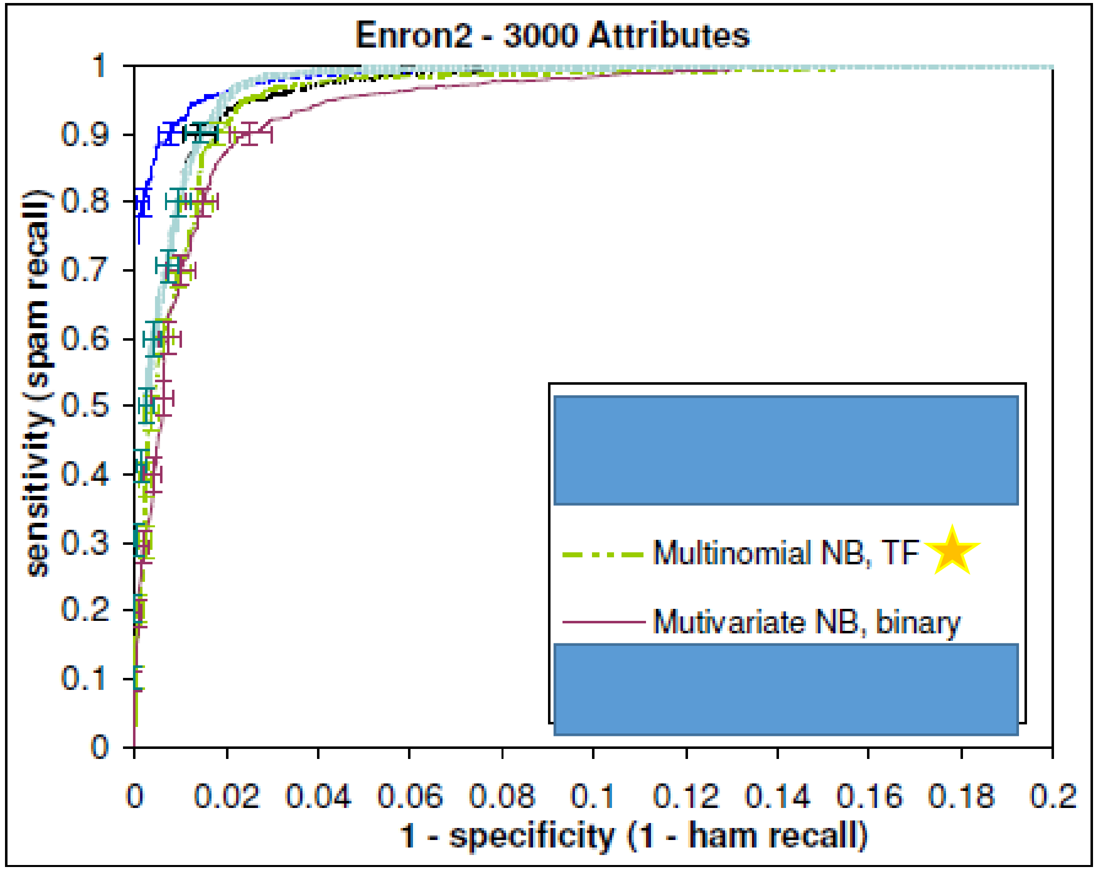
With Laplacian smoothing

$$p_{i,s} = \frac{1 + \#\text{ occurrences of term } i \text{ in spam}}{M + \sum_{i=1}^{M} \#\text{ occurrences of term } i \text{ in spam}}$$

Bernoulli NB Vs. Multinomial NB with TF

- Data for 6 different users from ENRON dataset
  - Augmented with spam emails from various sources (legit = “ham”)
  - Top-3000 features selected (we will discuss feature selection soon)

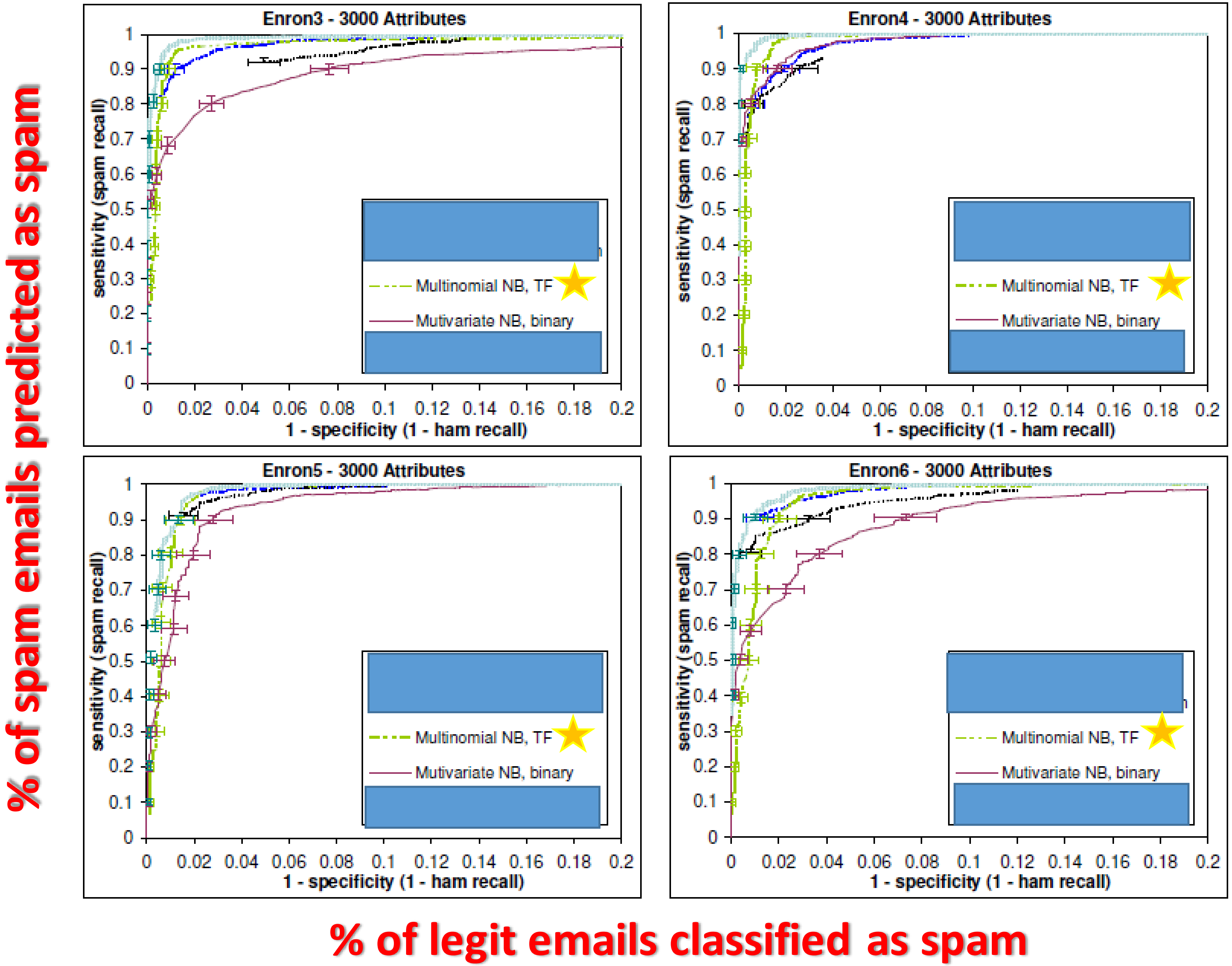




% of spam emails predicted as spam

% of legit emails classified as spam

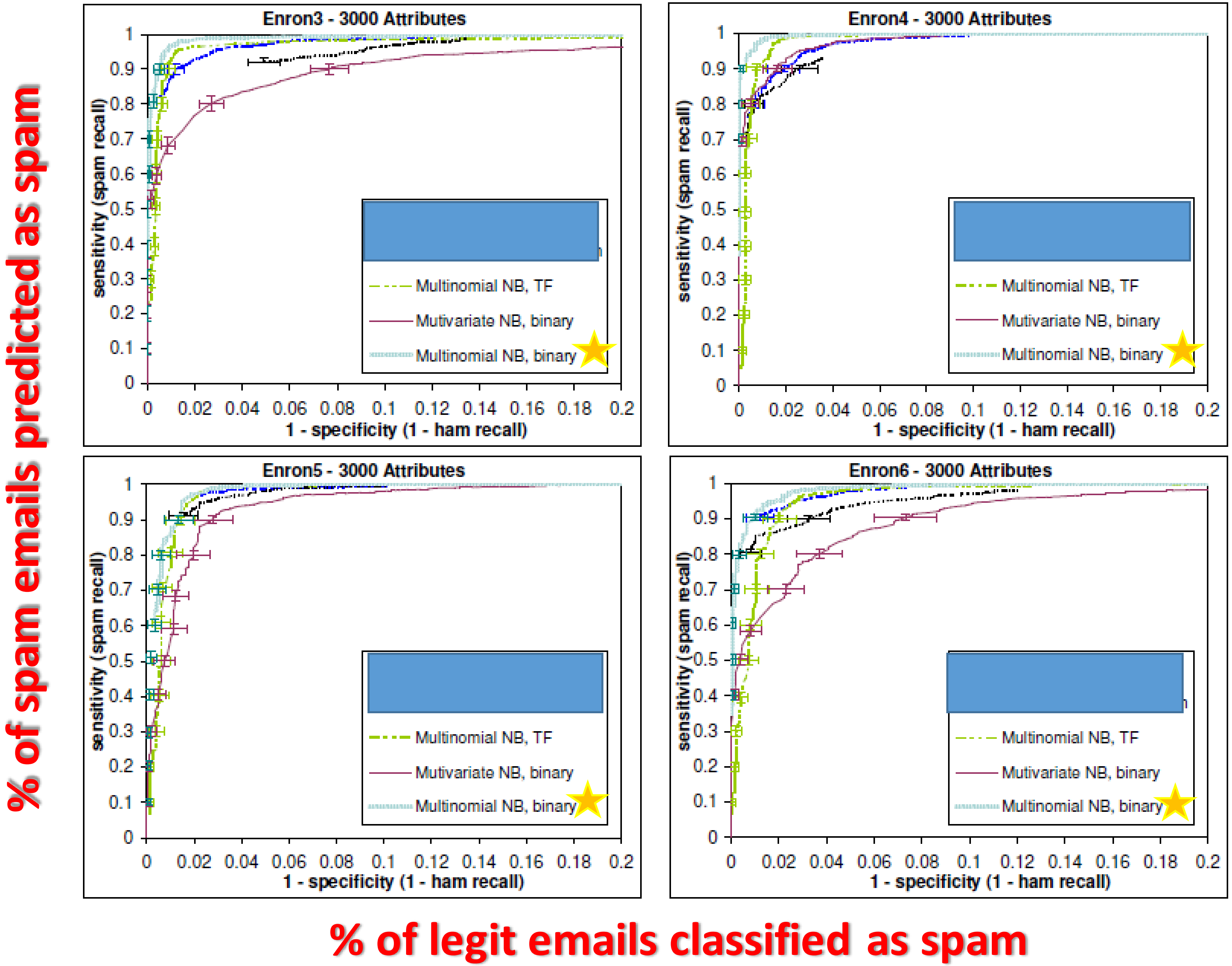
Bernoulli NB Vs. Multinomial NB with TF



Tempted to conclude that using term frequencies instead of binary occurrences helped in spam filtering.

Is there any other reason multinomial NB with TF might have outperformed Bernoulli NB?

Bernoulli NB Vs. Multinomial NB with TF



Multinomial NB with Binary instead of TF features performs the best!

Multinomial NB with Binary Features

- Let x = (x_1, x_2, …, x_M) be the TF features. Binary features are derived from the TF features as follows:

$$\overline{x} = (\overline{x_1} = \min(1, x_1), \overline{x_2} = \min(1, x_2), \ldots, \overline{x_M} = \min(1, x_M))$$

- Transformation is applied to both the training and test data and the multinomial model is used for prediction, i.e.,

$$P\{\overline{x} \mid spam\} = p(D)D!\prod_{i=1}^{M}\frac{(p_{i,s})^{\overline{x_i}}}{\overline{x_1}!} \qquad \left\{ \begin{array}{ll} p_{i,s} & \text{if } \overline{x_i} = 1 \\ 1 & \text{if } \overline{x_i} = 0 \end{array} \right.$$

Multinomial Vs. Bernoulli NB

Multinomial

$$P\{\overline{x} \mid spam\} = \cancel{p(D)} D! \prod_{i=1}^{M} (p_{i,s})^{\overline{x_i}}$$

Bernoulli

$$P\{x \mid spam\} = \prod_{i=1}^{M} p_{i,s}^{x_i} \, (1 - p_{i,s})^{1-x_i}$$

How are the two different?

1. Multinomial model *ignores negative evidence*
2. $p_{i,s}$ is estimated differently

$$\frac{1 + \#\text{ occurrences of term } i \text{ in spam}}{M + \sum_{i=1}^{M} \#\text{ occurrences of term } i \text{ in spam}}$$

$$\frac{1 + \#\text{Spam emails that contain term } i}{2 + \#\text{spam emails}}$$

Why Ignore Negative Evidence?

Schneider, Karl-Michael. "On word frequency information and negative evidence in Naive Bayes text classification." *Advances in Natural Language Processing*. Springer, Berlin, Heidelberg, 2004. 474-485.

Table 1. Statistics of the ling-spam corpus

	Total	Ling	Spam
Documents	2893	2412 (83.4%)	481 (16.6%)
Vocabulary	59,829	54,860 (91.7%)	11,250 (18.8%)

Vocabulary	Total		Ling		Spam	
	Words	Documents	Words	Documents	Words	Documents
Full	226.5	11.0	226.9	9.1	224.5	1.8
MI 5000	138.5	80.2	133.8	64.5	162.5	15.6
MI 500	44.0	254.5	39.6	190.9	66.2	63.7

Observation 1: >80% of words never occur in spam documents, while only 10% of words never occur in legit documents

Observation 2: On average, documents only contain a very small fraction of words from the vocabulary

For Bernoulli NB, probability of a document is mostly determined by words that <u>do not</u> appear in the document!

Why is Multinomial Binary Features better than Term Frequencies?

Multinomial TF assumes repeated instances of the same word occur independently, but that is not the case -> for example, if a word appears once it is more likely to appear multiple times. Therefore multinomial TF is a poor model for the underlying data.

Spam Filtering Review

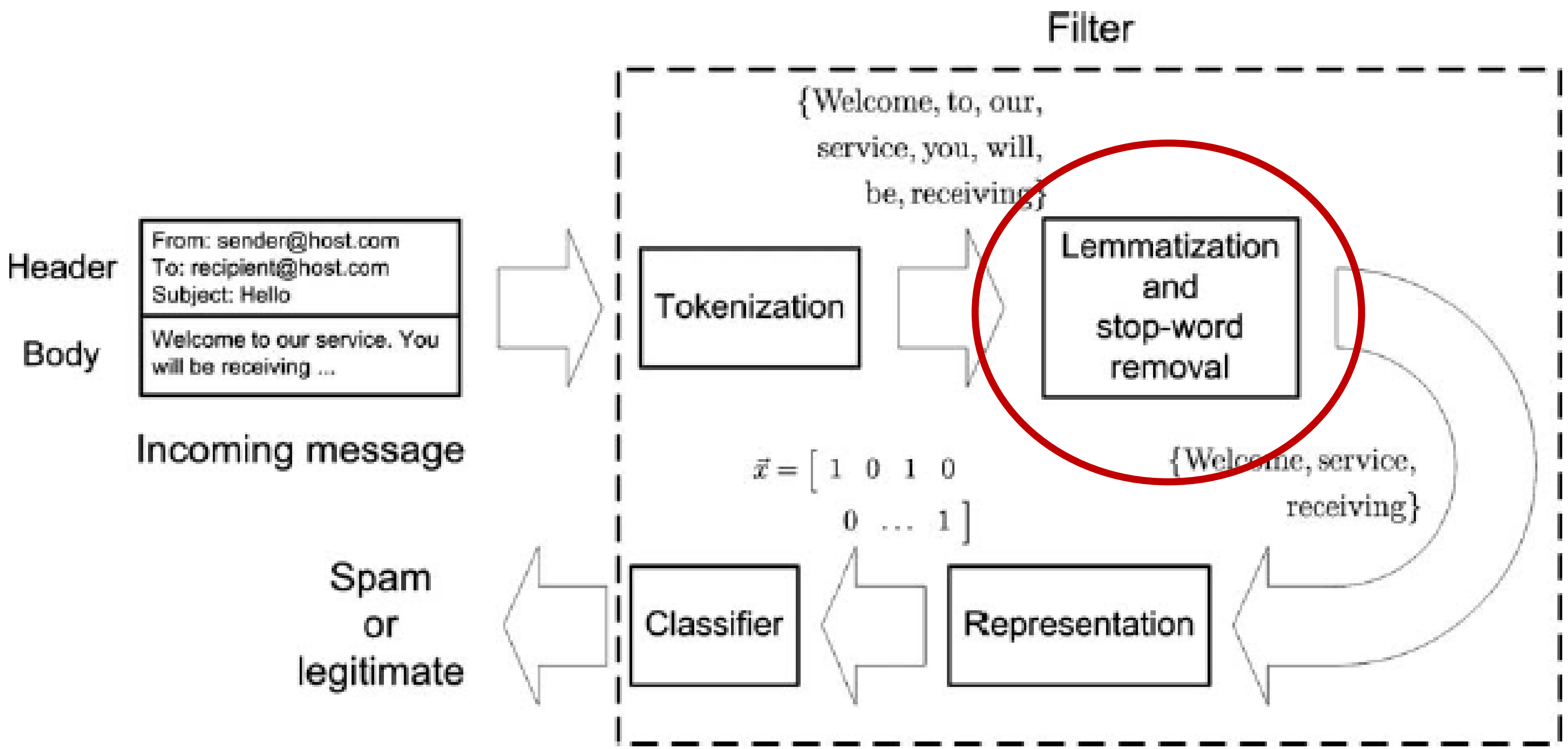


Fig. 1. An illustration of some of the main steps involved in a spam filter.

Lemmatization

Subject: re : 2 . 882 s - > np np> date : sun , 15 dec 91 02 : 25 : 02 est > from : michael < mmorse @ vm1 . yorku . ca > > subject : re : 2 . 864 queries > > wlodek zadrozny asks if there is " anything interesting " to be said > about the construction " s > np np " . . . second , > and very much related : might we consider the construction to be a form > of what has been discussed on this list of late as reduplication ? the > logical sense of " john mcnamara the name " is tautologous and thus , at > that level , indistinguishable from " well , well now , what have we here ? " . to say that ' john mcnamara the name ' is tautologous is to give support to those who say that a logic-based semantics is irrelevant to natural language . in what sense is it tautologous ? it supplies the value of an attribute followed by the attribute of which it is the value . if in fact the value of the name-attribute for the relevant entity were ' chaim shmendrik ' , ' john mcnamara the name ' would be false . no tautology , this . (and no reduplication , either .)

AFTER LEMMATIZING

Subject: re : 2 . 882 s - > np np> deat : sun , 15 dec 91 2 : 25 : 2 est > from : michael < mmorse @ vm1 . yorku . ca > > subject : re : 2 . 864 query > > wlodek zadrozny ask if there be " anything interest " to be say > about the construction " s > np np " . . . second , > and very much relate : may we consider the construction to be a form > of what have be discuss on this list of late as reduplication ? the > logical sense of " john mcnamara the name " be tautologous and thus , at > that level , indistinguishable from " well , well now , what have we here ? " . to say that ' john mcnamara the name ' be tautologous be to give support to those who say that a logic-base semantics be irrelevant to natural language . in what sense be it tautologous ? it supplies the value of an attribute follow by the attribute of which it be the value . if in fact the value of the name-attribute for the relevant entity be ' chaim shmendrik ' , ' john mcnamara the name ' would be false . no tautology , this . (and no reduplication , either .)

Stop-Words

Subject: re : 2 . 882 s - > np np> date : sun , 15 dec 91 02 : 25 : 02 est > from : michael < mmorse @ vm1 . yorku . ca > > subject : re : 2 . 864 queries > > wlodek zadrozny asks if there is " anything interesting " to be said > about the construction " s > np np " . . . second , > and very much related : might we consider the construction to be a form > of what has been discussed on this list of late as reduplication ? the > logical sense of " john mcnamara the name " is tautologous and thus , at > that level , indistinguishable from " well , well now , what have we here ? " . to say that ' john mcnamara the name ' is tautologous is to give support to those who say that a logic-based semantics is irrelevant to natural language . in what sense is it tautologous ? it supplies the value of an attribute followed by the attribute of which it is the value . if in fact the value of the name-attribute for the relevant entity were ' chaim shmendrik ' , ' john mcnamara the name ' would be false . no tautology , this . (and no reduplication , either .)

AFTER STOP WORDS

Subject: re : 2 . 882 s - > np np> date : sun , 15 dec 91 02 : 25 : 02 est > : michael < mmorse @ vm1 . yorku . ca > > subject : re : 2 . 864 queries > > wlodek zadrozny asks is " anything interesting " said > construction " s > np np " . . . second , > much related : might consider construction form > has been discussed list late reduplication ? > logical sense " john mcnamara name " is tautologous thus , > level , indistinguishable " , , here ? " . ' john mcnamara name ' is tautologous is support those logic-based semantics is irrelevant natural language . sense is tautologous ? supplies value attribute followed attribute is value . fact value name-attribute relevant entity were ' chaim shmendrik ' , ' john mcnamara name ' false . tautology , . (reduplication , either .)

Feature Selection

- Feature space of text classification problems can be large
  - Size of the vocabulary in the worst-case
    - Increases the **computational costs** of training a model and performing predictions
    - **Model complexity**?
- Goal: reduce the size of the feature space by retaining only the top-N features
  - Example: TF of Top-N terms that help predict whether a message is spam
  - But how do we select features
    - Fix an N and try all subsets of N features?

Feature Selection

Yang, Yiming, and Jan O. Pedersen. "A comparative study on feature selection in text categorization." *Icml*. Vol. 97. 1997.

- Features are selected based on statistical or information-theoretic metrics that rank terms in order of discriminative power
  - Document frequency
  - Information Gain (IG)
  - Mutual Information (MI)
  - χ^2 Statistic
  - Term Importance (TI)

- Document Frequency: retain only the top-N most frequently occurring term in the training dataset
  - What about infrequent/rare but highly informative terms?
  - Common but non-informative terms? Arguably stop-lists are doing the opposite

Information Gain (IG)

- IG measures the "number of bits of information the presence or absence of a term reveals about the document category (spam/legit)"

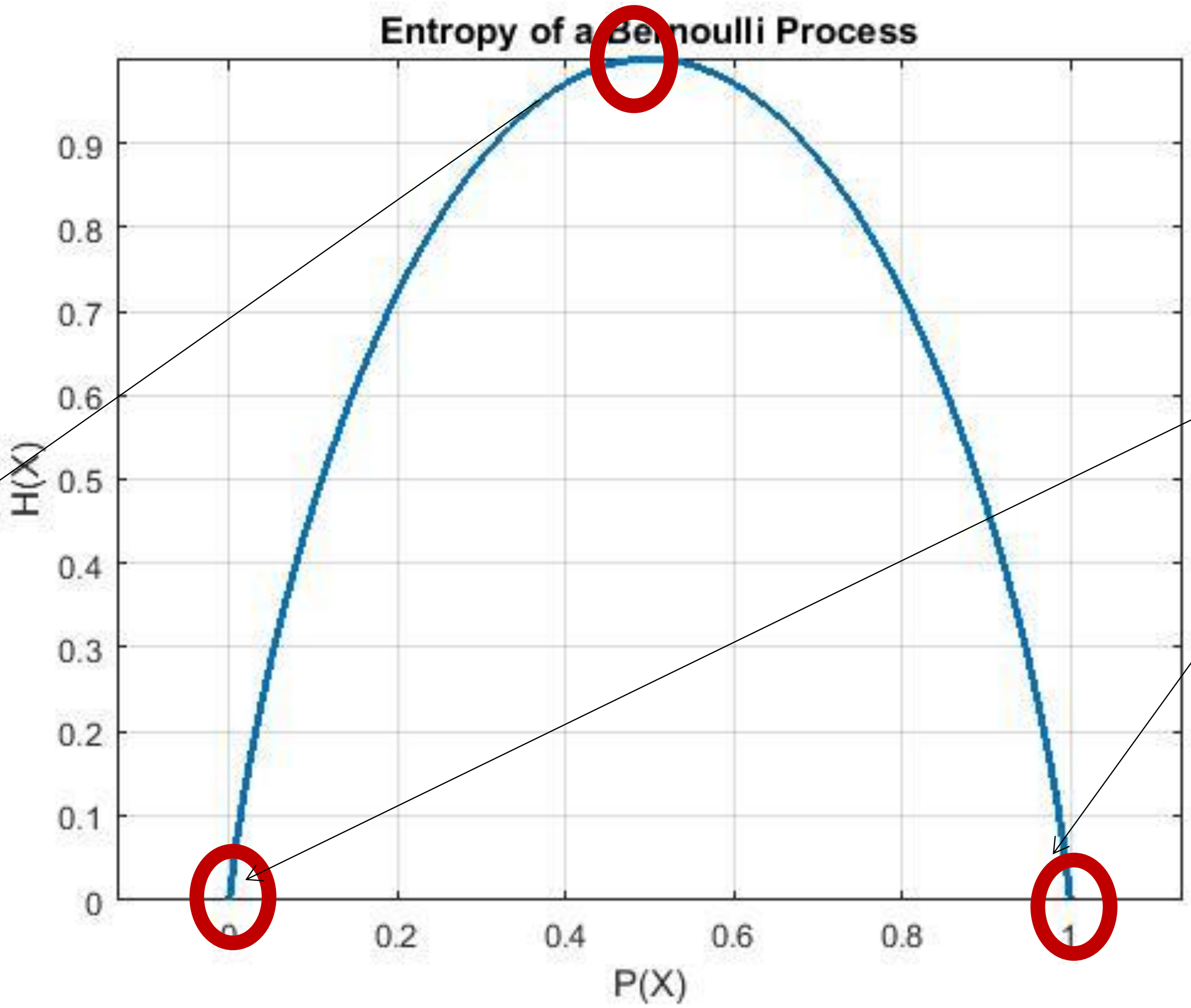
- But how do we measure "information"
  - **Entropy**: the entropy of a random variable *X* is

$$H(X) = -\sum_{x} p(X = x)\log(P(X = x))$$ Measured in "bits"

  - Consider a Bernoulli random variable $X \in \{0,1\}$ and assume that $p(X = 1) = p$
  - What is H(*X*)?

$$H(X) = -p\log(p) - (1 - p)\log(1 - p)$$

Entropy of Bernoulli RV



Entropy = 0 bits
Outcome of RV reveals no information that you didn't already have!

Entropy = 1 bit
Each trial reveals one full bit of information

Back to Information Gain

1. Let C be a RV that determines if a document is spam or legit
   - H(C) is the inherent uncertainty in the RV
2. Let X_i be a RV that represents the occurrence of frequency of term i
   - Can be either binary or TF

How much information does X_i provide about C

IG measures the reduction in entropy of C if X_i is known

$$IG(C, X_i) = H(C) - H(C \mid X_i)$$

Inherent uncertainty

Uncertainty given X_i

Conditional Entropy

$$\boxed{IG(C, X_i) = H(C) - H(C \mid X_i)}$$

$$H(C \mid X_i) = \sum_x P(X_i = x) H(C \mid X = x)$$

$$= \sum_{x,c} P(X_i = x, C = c) \log(P(C = x \mid X_i = x))$$

Assuming binary features:

$$H(C \mid X_i) = -\sum_{c \in \{spam, legit\}} \sum_{x \in \{0,1\}} P(X_i = x, C = c) \log(P(C = x \mid X_i = x))$$

$$P(X_i = x \mid C = c) * P(C = c)$$

$$\frac{P(X_i = x \mid C = c) * P(C = c)}{P(X_i = x)}$$

In-Class Exercise

\# Spam Emails in Training Dataset: 50

\# Legit Emails in Training Dataset: 100

Word/Term	#Spam Emails with Term	#Legit Emails with Term
“FREE”	40	0
“George”	0	20
“and”	40	80

Compute the IG of “Free”, “George” and “and”

χ^2 Test Statistic

- χ^2 test is a commonly used statistical test to measure the independence between two random variables

$$\chi^2(C=c, X_i) = \frac{N*(AD-BC)}{(A+C)*(B+D)+(A+B)*(C+D)}$$

A: Number of instances in which document class c and X_i co-occur
B: Number of instance in which term X_i occurs in other document classes
C: Number of documents of class c that don't have term X_i
D: Number of instances of other "non-c" document classes that don't have term X_i

- If there are only two classes then $\chi^2(C=c, X_i) = -\chi^2(C=\bar{c}, X_i)$ so we use

$$|\chi^2(C=c, X_i)|$$

In-Class Exercise

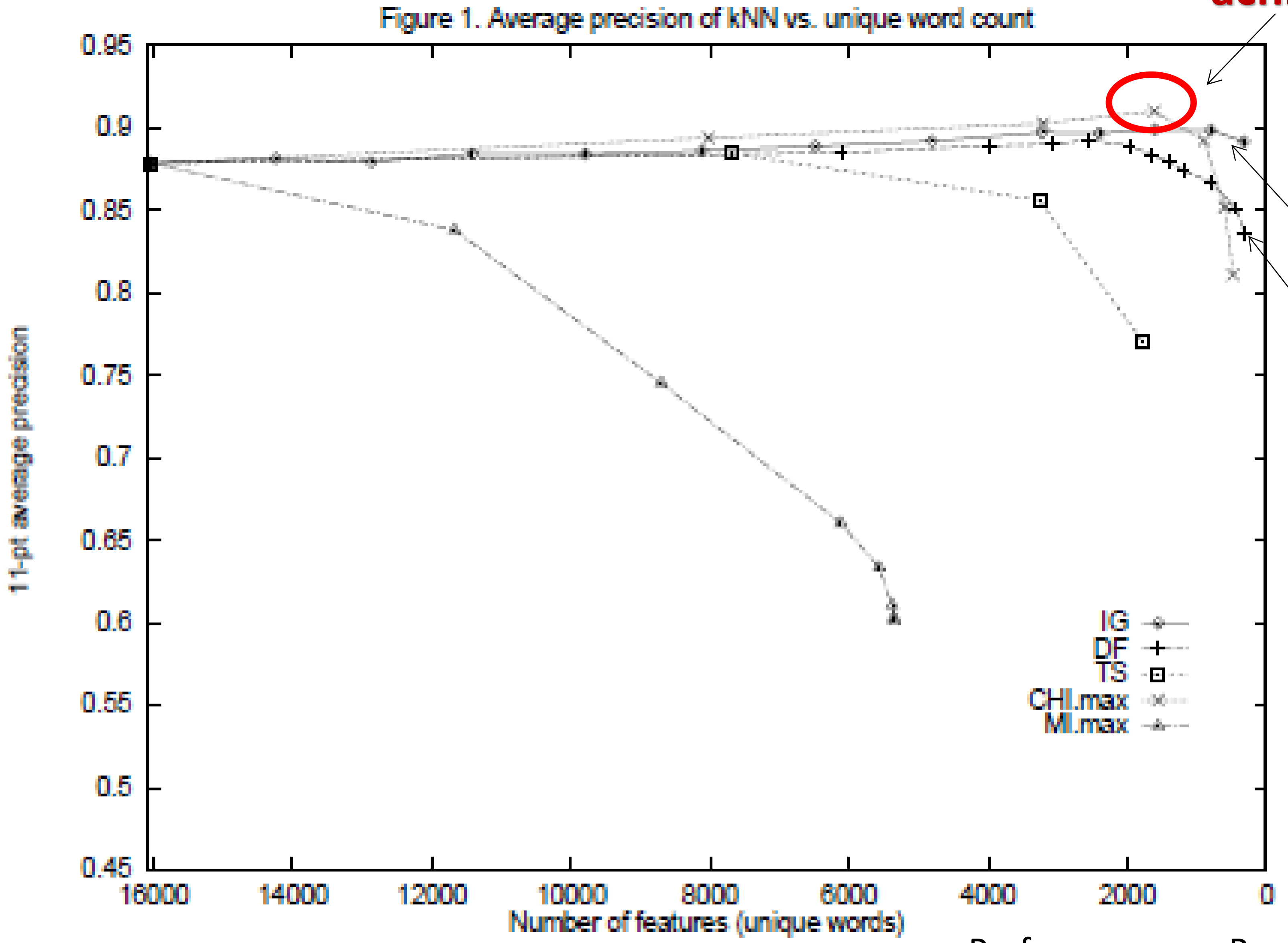
Spam Emails in Training Dataset: 50

Legit Emails in Training Dataset: 100

Word/Term	#Spam Emails with Term	#Legit Emails with Term
“FREE”	40	0
“George”	0	20
“and”	40	80

Compute the χ^2 statistic of “Free”, “George” and “and”

Empirical Results



Maximum precision achieved by χ^2

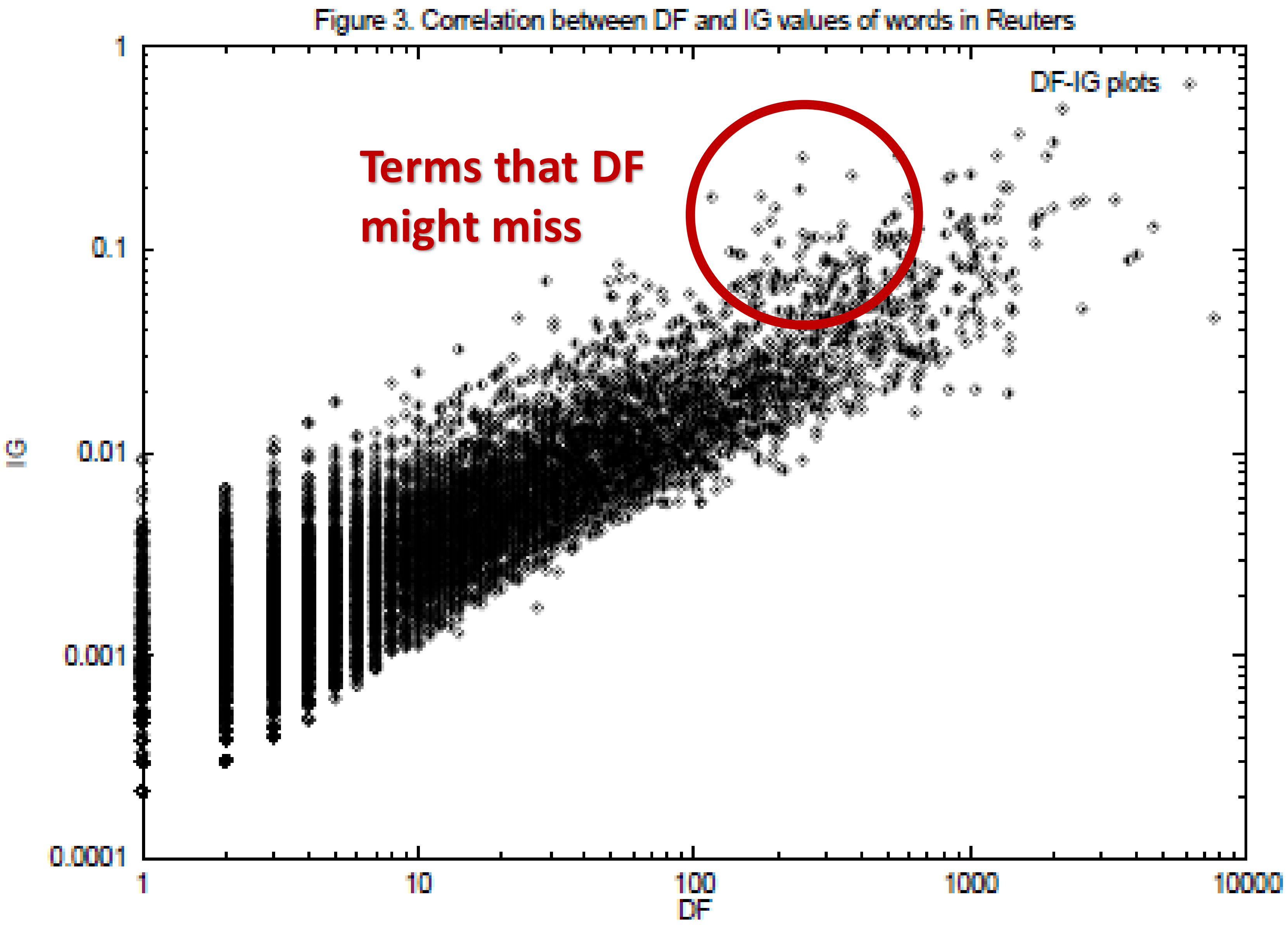
IG performs best with very few features

IG

DF has least precision but still competitive, especially as number of features increases

Performance on Reuters dataset with kNN classfier

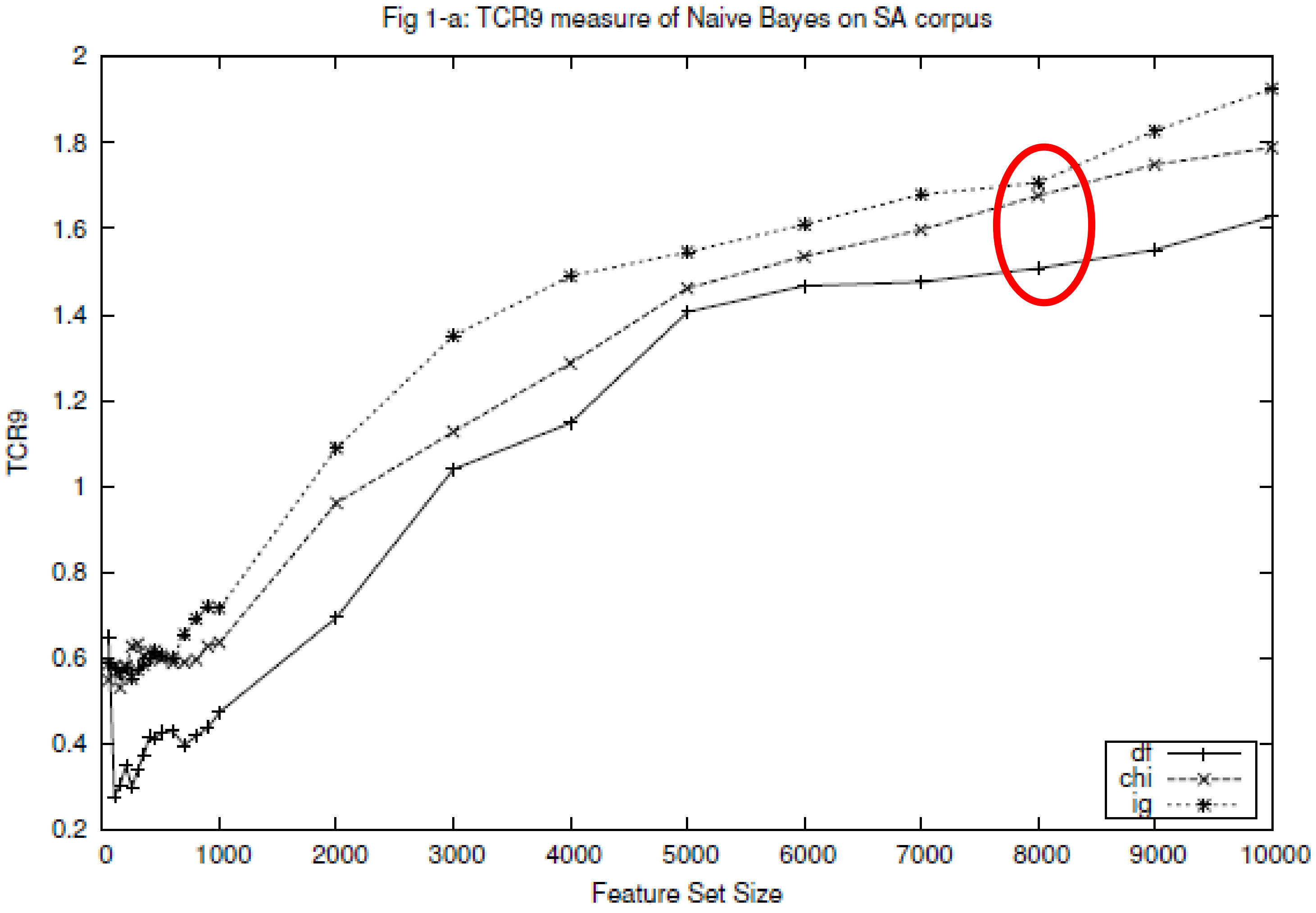
DF Vs. IG



Performance on Reuters dataset with kNN classfier

Results on Spam Dataset

Zhang, Le, Jingbo Zhu, and Tianshun Yao. "An evaluation of statistical spam filtering techniques." *ACM Transactions on Asian Language Information Processing (TALIP)* 3.4 (2004): 243-269.



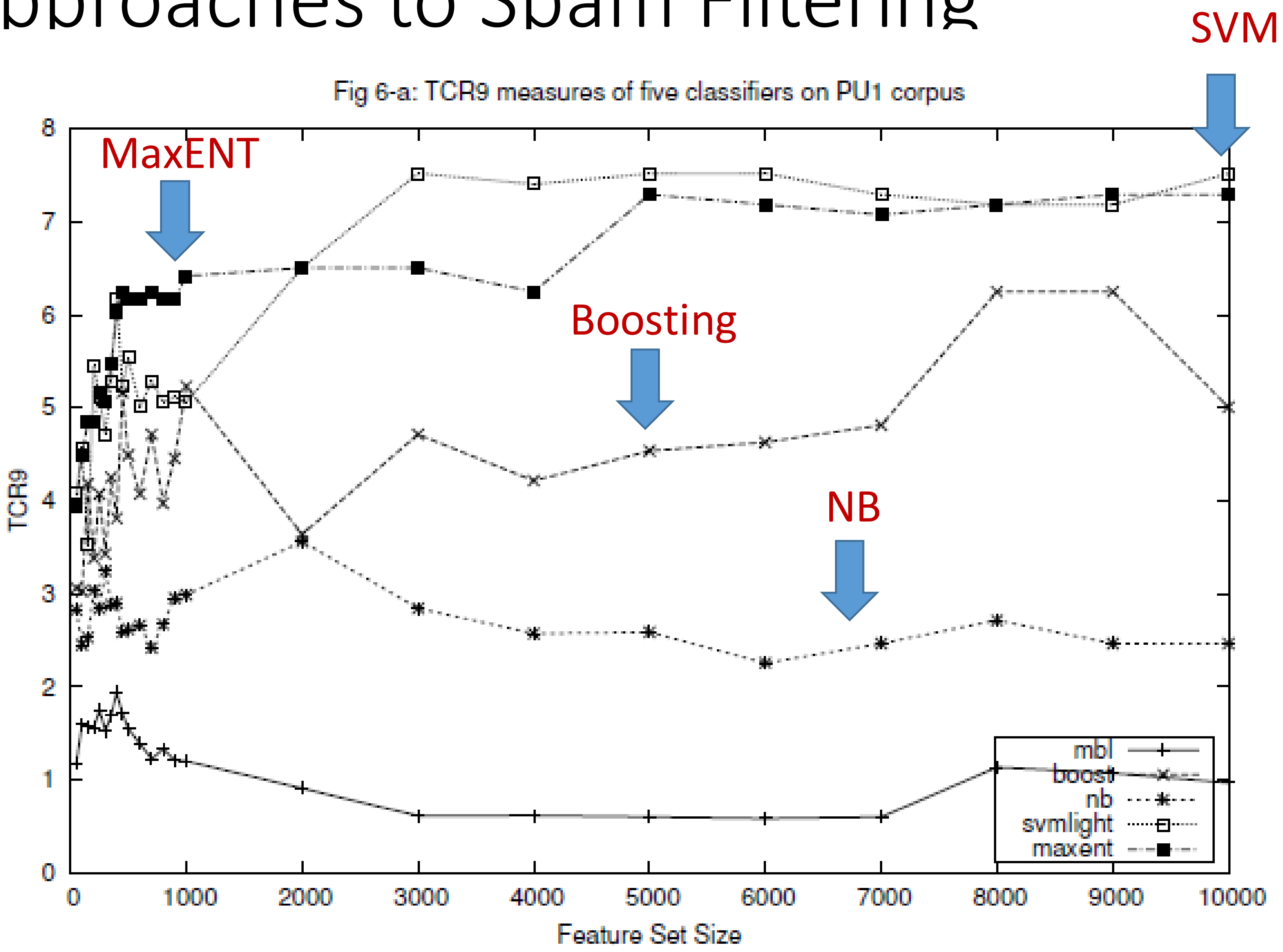
$$TCR = \frac{WErr^b}{WErr} = \frac{N_S}{\lambda \cdot n_{L\to S} + n_{S\to L}}$$

So What's Used in Literature

Table 1
Some of the most common feature selection methods applied in Spam filtering.

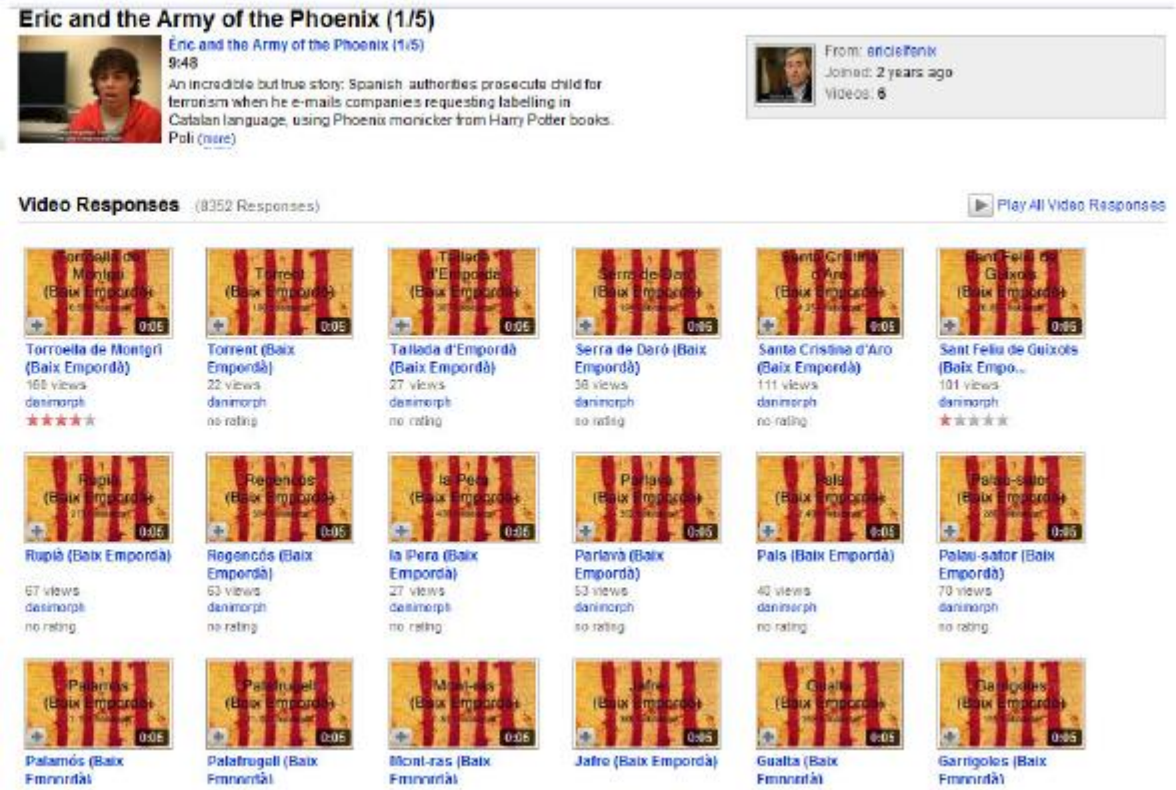
Name	Term score	Number of works
Document frequency	$\tau(t_i) = \lvert\{d : d \in \mathcal{D}_{tr} \text{ and } t_i \in d\}\rvert$	2
Information gain	$\tau(t_i) = \sum_{c\in\{s,l\}} \sum_{t\in\{t_i,\bar{t}_i\}} P(t,c) \log\left[\frac{P(t,c)}{P(t)P(c)}\right]$	26
χ^2 statistic	$\tau(t_i,c) = \frac{\lvert\mathcal{D}_{tr}\rvert (P(t_i,c)P(\bar{t}_i,\bar{c}) - P(\bar{t}_i,c)P(t_i,\bar{c}))^2}{P(t_i)P(\bar{t}_i)P(c)P(\bar{c})}$	1
Odds ratio	$\tau(t_i,c) = \frac{P(t_i\|c)}{1-P(t_i\|c)} \frac{1-P(t_i\|\bar{c})}{P(t_i\|\bar{c})}$	1
Term-frequency variance	$\tau(t_i) = \sum_{c\in\{s,l\}} (T_f(t_i,c) - T_f^{\mu}(t_i))^2$	2

Other Approaches to Spam Filtering



Spam Detection on Social Media

Benevenuto, Fabrício, et al. "Detecting spammers and content promoters in online video social networks." *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2009.



- **Spammers** try to poison search results in order to get more views for unrelated videos
- **Promoters** post unrelated videos to increase the relevance of certain topics

What are the Right Set of Features

- Since we're looking at social networks, it might be meaningful to exploit social network structure

- How do we capture the structure of a graph as a number?

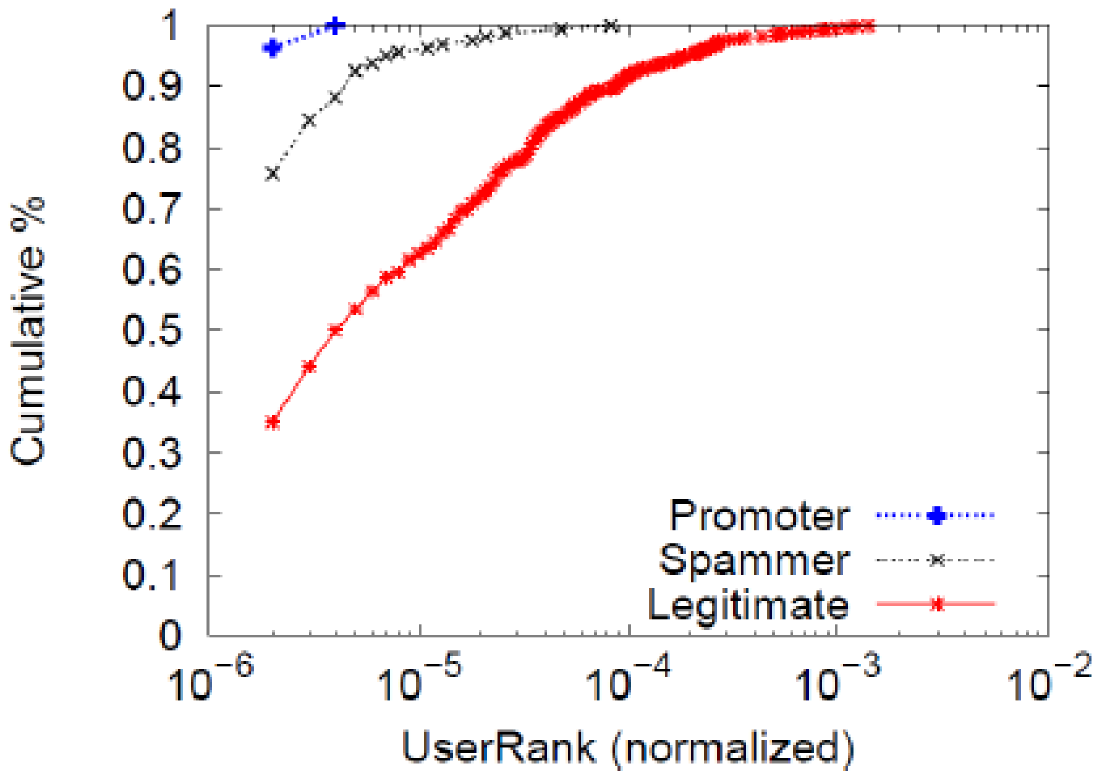
Features/Attributes Used

- Social Network features
  - Clustering coefficient, "betweenness", UserRank etc. (more on this later)
- User-based
  - Number of friends, number of subscriptions, number of subscribers
- Video-based
  - Duration, number of views, number of comments received, ratings

Feature Selection: χ^2 ranking

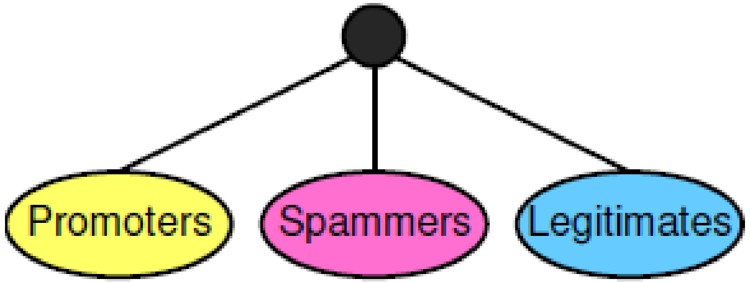
Attribute Set	Top 10	Top 20	Top 30	Top 40	Top 50
Video	9	18	25	30	36
User	1	2	4	7	9
SN	0	0	1	3	5

UserRank as a Feature



Even low-ranked features have potential
to separate classes apart

Classification Results Using SVMs



- Correctly identify majority of promoters, misclassifying a small fraction of legitimate users.
- Detect a significant fraction of spammers but they are much harder to distinguish from legitimate users.
  - Dual behavior of some spammers

		Predicted		
		Promoter	Spammer	Legitimate
	Promoter	**96.13%**	3.87%	0.00%
True	Spammer	1.40%	**56.69%**	41.91%
	Legitimate	0.31%	5.02%	**94.66%**

- Micro F1 = 88% (predict the correct class 88% of cases)