Lecture 4: Adversarial Attacks on Spam Filter

Siddharth Garg
sg175@nyu.edu

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 ocurring 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)"

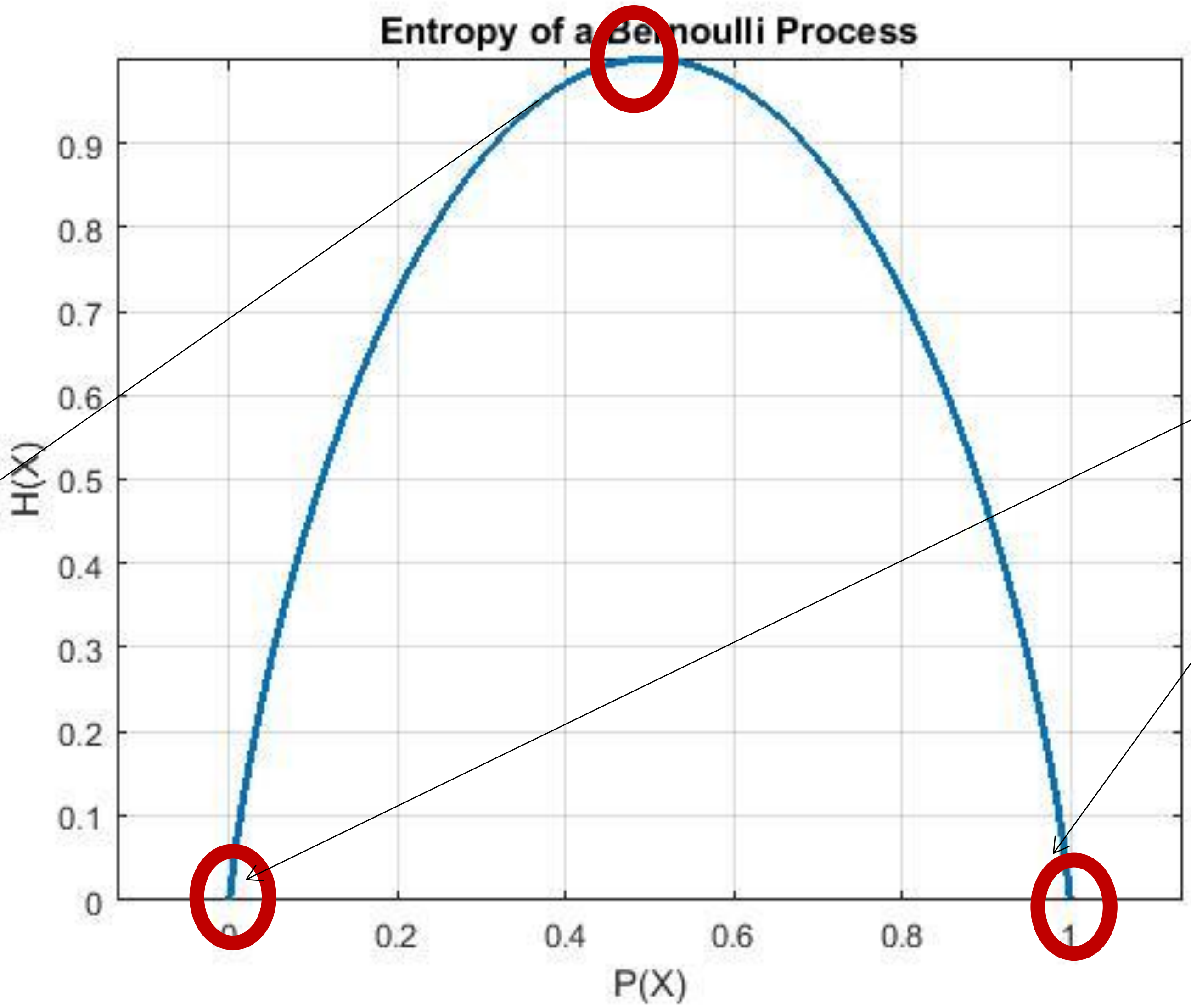
- But how do we measure "information"
  - **Entropy**: the entropy of a random variable *X* is

$$H(X) = -\sum_{x} p(X = x)\log(P(X = x))$$ 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

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

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 "non-c" 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

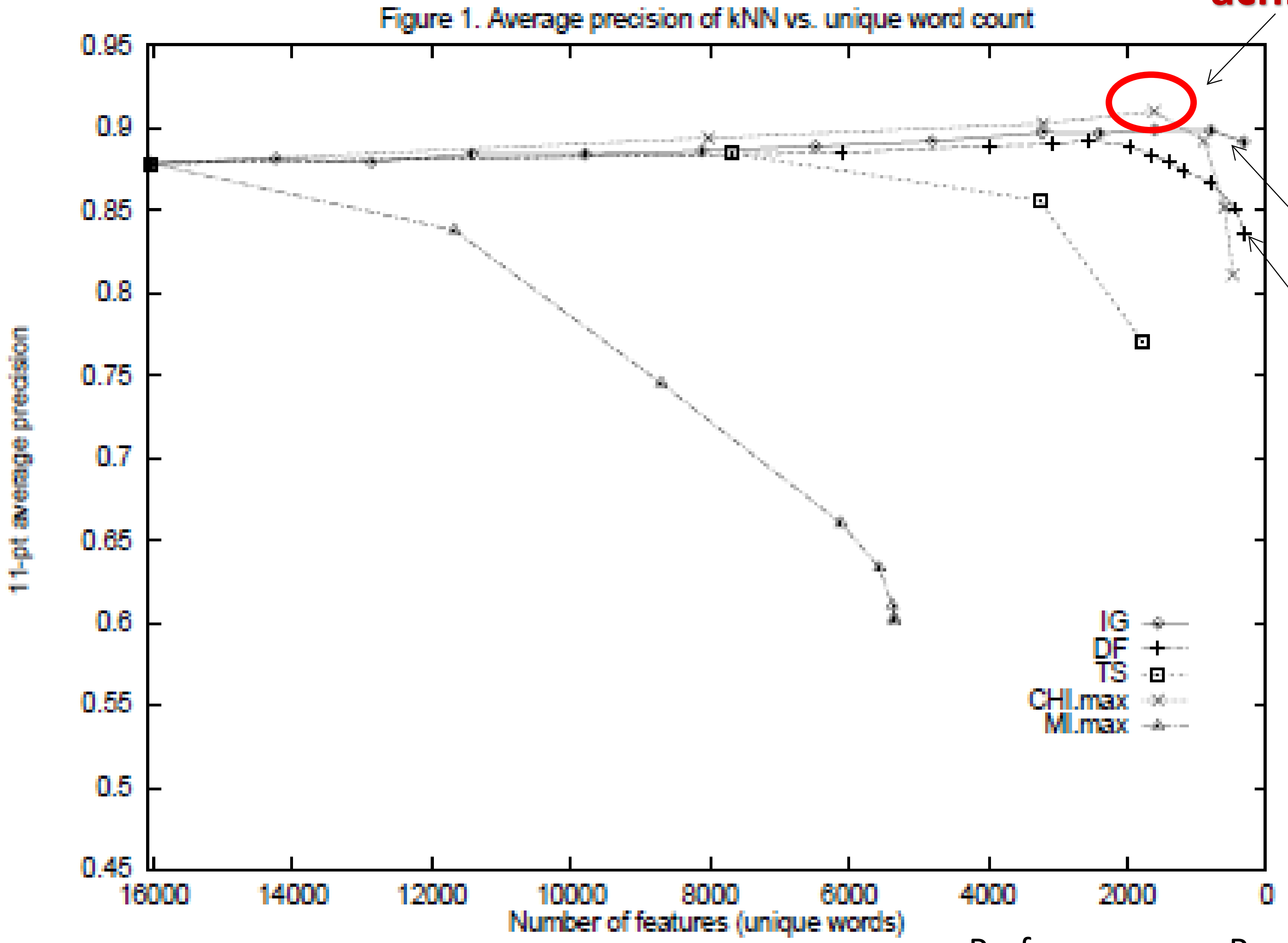
Spam Emails in Training Dataset: 50

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

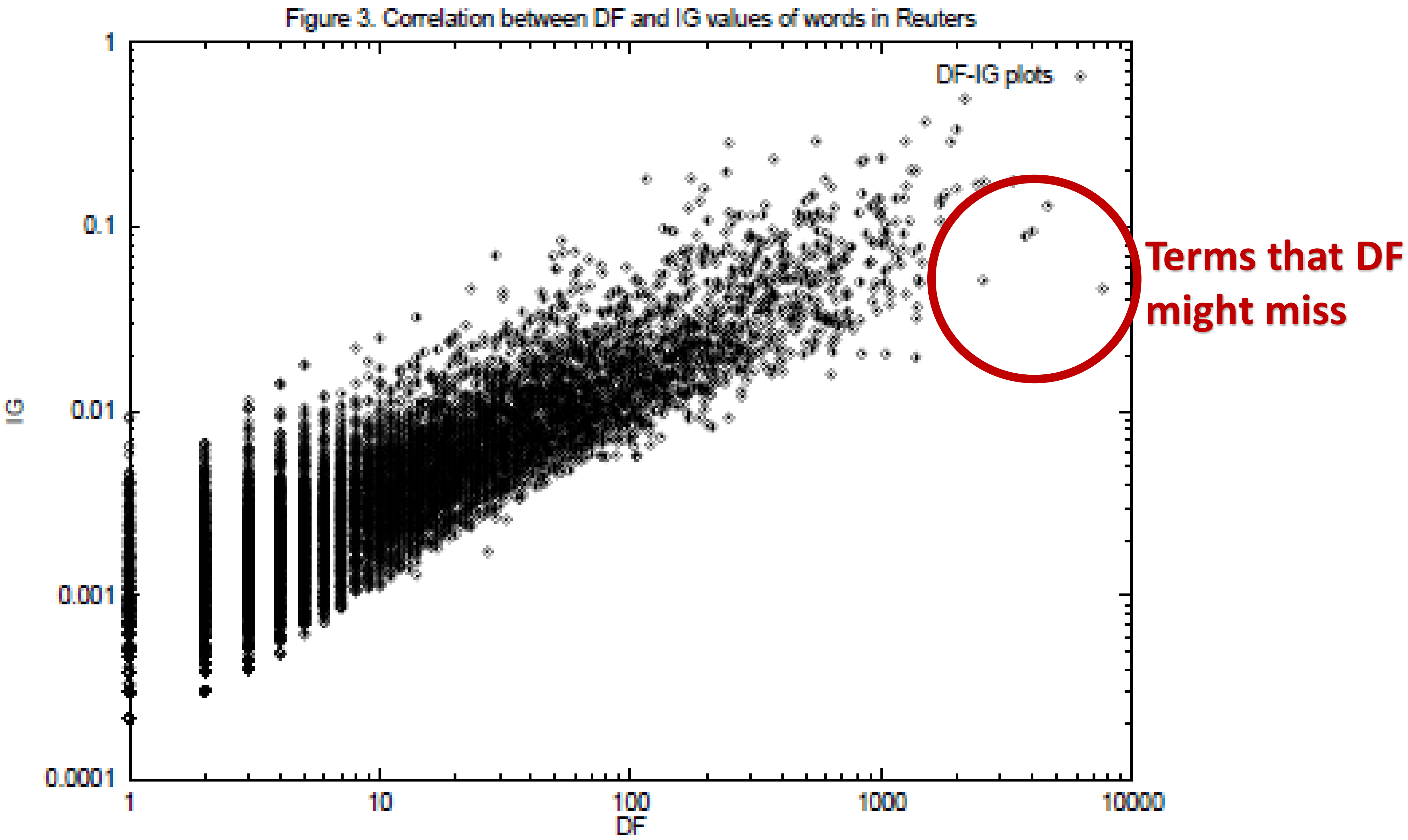
IG performs best with very few features

IG

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

Performance on Reuters dataset with kNN classfier

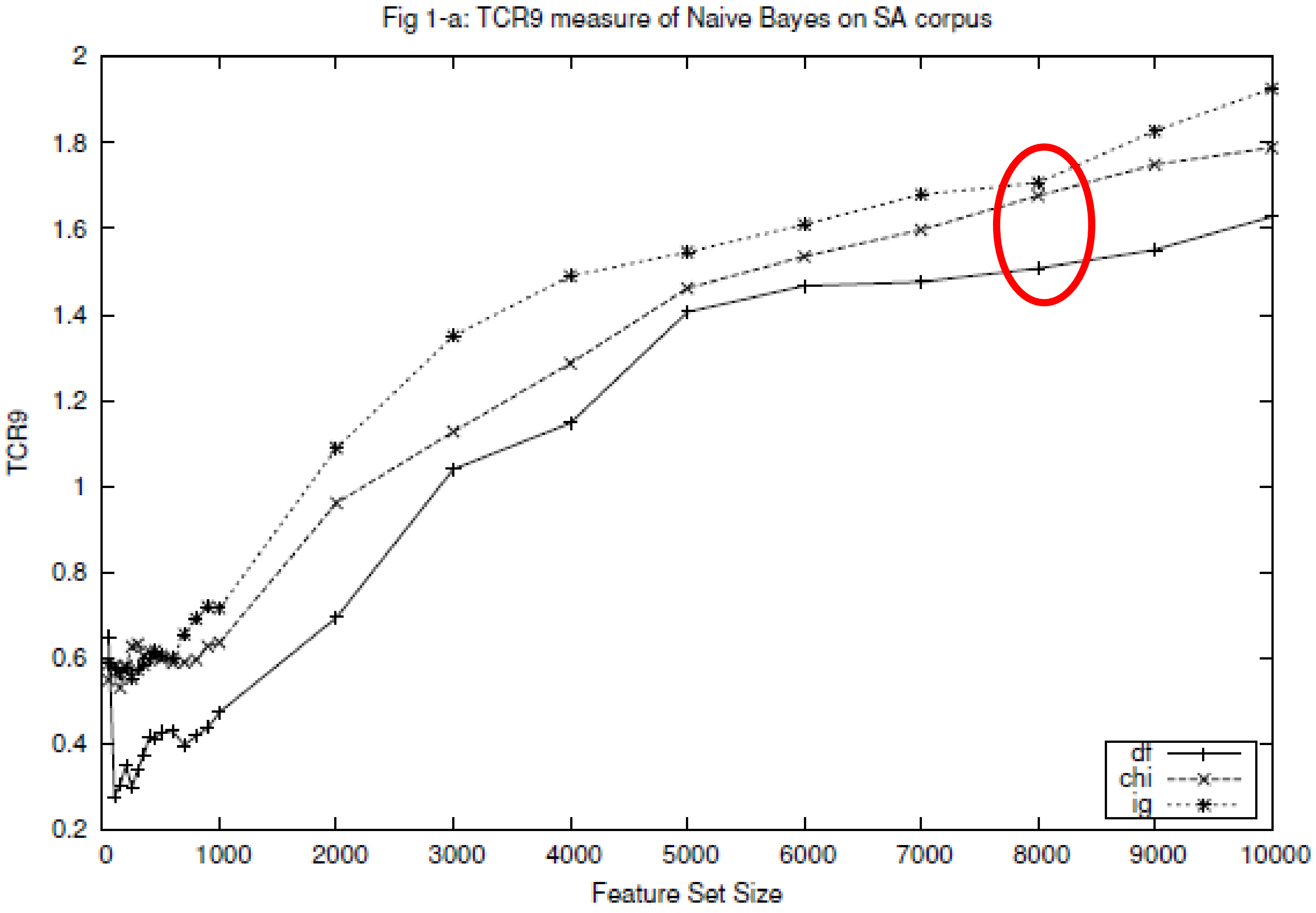
DF Vs. IG



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.



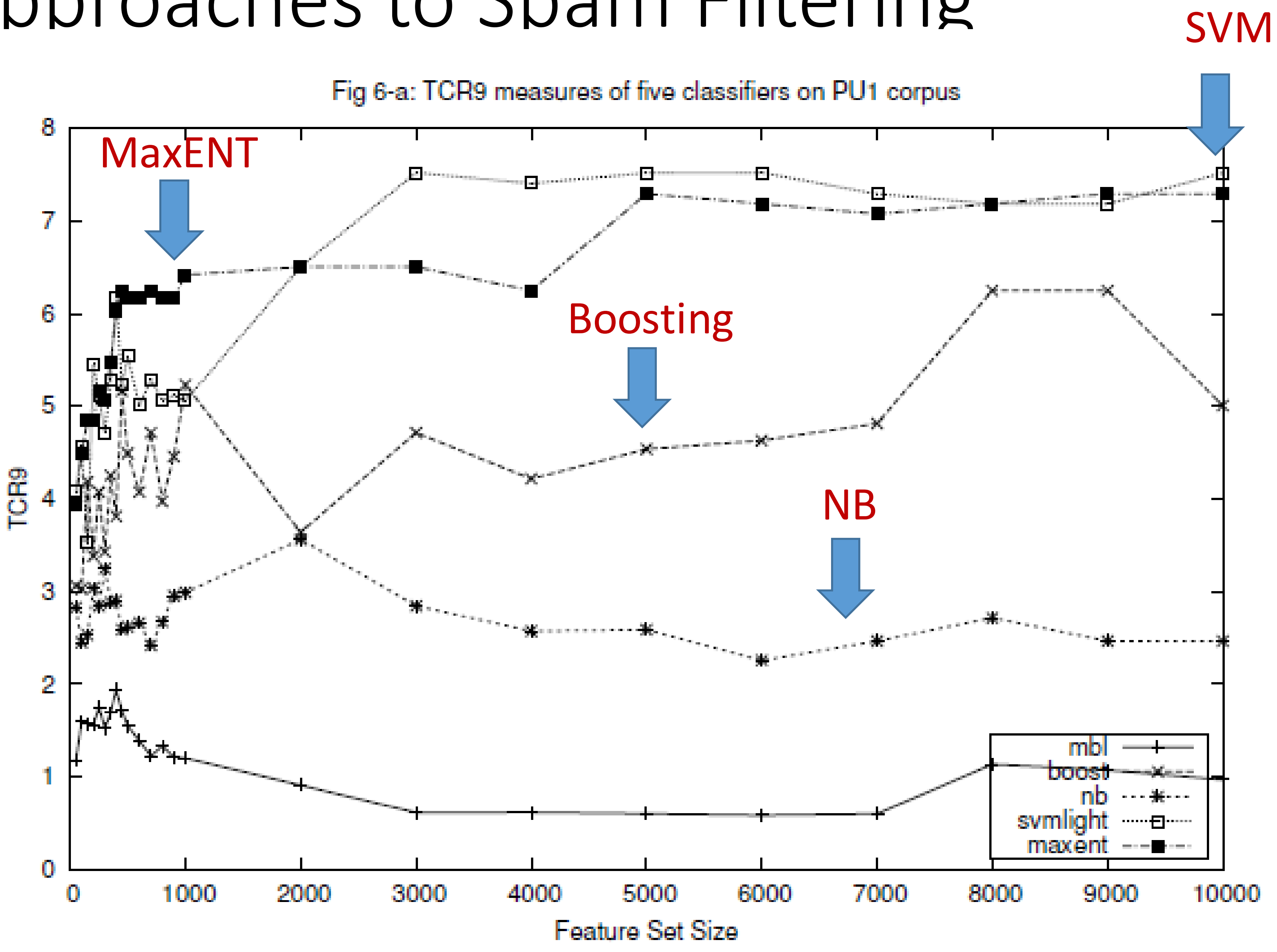
$$TCR = \frac{WErr^b}{WErr} = \frac{N_S}{\lambda \cdot n_{L\to S} + n_{S\to L}}$$

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	$\tau(t_i,c) = \frac{P(t_i\mid c)}{1-P(t_i\mid c)}\frac{1-P(t_i\mid\bar{c})}{P(t_i\mid\bar{c})}$	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



How Do Adversaries Respond to ML Defenses

- Real-world adversaries are adaptive, i.e., they attempt to evade defenses using smarter attacks

From: spammer@example.com
Cheap mortgage now!!

Feature Weights

cheap = 1.0
mortgage = 1.5

Total score = 2.5 > 1.0 (threshold)

Spam

D. Lowd, "Adversarial Statistical Relational AI"

Spammers Adapt

1. From: spammer@example.com
Cheap mortgage now!!
Eugene Oregon

Feature Weights

2. cheap = 1.0
mortgage = 1.5
Eugene = -1.0
Oregon = -1.0

3. Total score = 0.5 < 1.0 (threshold)

OK

D. Lowd, "Adversarial Statistical Relational AI"

Classifier Adapts

From: spammer@example.com
Cheap mortgage now!!
Eugene Oregon

Feature Weights

cheap = 1.5
mortgage = 2.0
Eugene = -0.5
Oregon = -0.5
Total score = 2.5 > 1.0 (threshold)

Spam

D. Lowd, “Adversarial Statistical Relational AI”

Attacker’s Strategy

- Recall that the NB classifier computes and compares

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

- Attacker seeks to change x to x’ such that $P\{spam \mid x'\} > P\{spam \mid x\}$ which also implies that $P\{legit \mid x'\} < P\{legit \mid x\}$

Attacker’s Strategy for Bernoulli NB

- Recall that for Bernoulli NB

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

where $x_i \in \{0,1\}$

- Assume an email *x* such that $P\{spam \mid x\} > P\{legit \mid x\}$ and $x_i = 0$
  - **Under what condition does adding term *i* to the document help the attacker?**

Attacker's Strategy for Multinomial NB

- Recall that for Bernoulli NB

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

where $x_i \in \{0,1\}$

- Assume an email *x* such that $P\{spam \mid x\} > P\{legit \mid x\}$ and $x_i = 0$
  - **Under what condition does adding term *i* to the document help the attacker?**

Attacker's Strategy for Multinomial NB

- Recall that for Bernoulli NB

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

where $x_i \in \aleph$

- Assume an email x such that $P\{spam \mid x\} > P\{legit \mid x\}$ and $x_i = r$
    - **Should the attacker increase or decrease the number of instances of term *i*?**

“Game” Between Attacker and Defender

Notion of cost?

Designs classifier C(x) to max. classification accuracy

C(x) and it’s parameters

Determines a function x’ = A(x) such that for each x where C(x)=spam , C(x’) = legit

A(x) and i

param

Designs new classifier C’ to max. accuracy of C’(A(x))

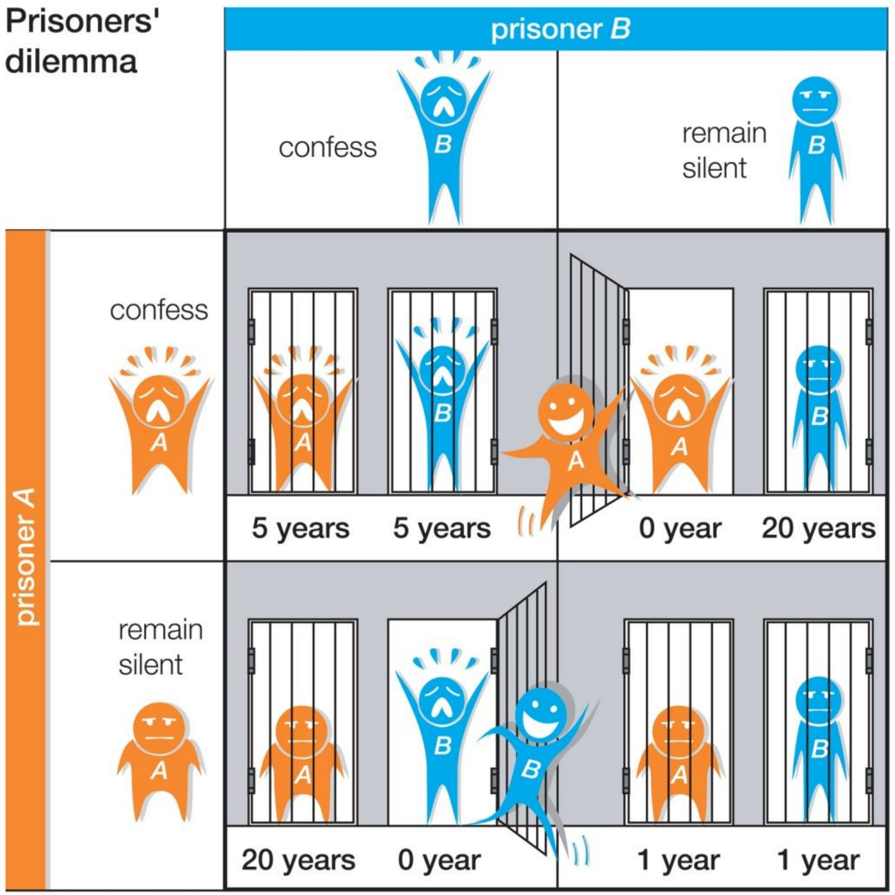
x) and it’s parameters

Determines a function x’ = A’(x) such that for each x where C’(x)=spam , C’(x’) = legit

A’(x) and it’s parameters

When Does it End

Game Theoretic Analysis



© 2010 Encyclopædia Britannica, Inc.

- What should the prisoner's strategies be?
  - Assume full information, i.e., both prisoner's know each other's costs/utilities
  - Nash Equilibrium: a pair of strategies such that neither prisoner has incentive to *unilaterally* deviate
  - What is the NE for this game?

WORSE OUTCOME THAN IF THEY BOTH REMAINED SILENT!

http://www.acting-man.com/blog/media/2014/11/prisoners_dilemma.jpg

Nash Equilibrium for Email Game

Classifier *C* is optimal keeping in mind the adversary's camouflaging strategy A

Classifier C and its parameters

"Camouflager" A and its parameters

Camouflager *A* maximizes the adversary's utility given classifier *C.* Adversary's utility accounts for spam emails that get classified as legit and "cost" of modifications

Does a Nash Equilibrium exist? Can it be efficiently computed?

“Game” Between Attacker and Defender

Notion of cost?

Designs classifier C(x) to max. classification accuracy

C(x) and it’s parameters

Determines a function x’ = A(x) such that for each x where C(x)=spam , C(x’) = legit

A(x) and it’s parameters

Designs new classifier C’ to max. accuracy of C’(A(x))

C’(x) and it’s parameters

Determines a function x’ = A’(x) such that for each x where C’(x)=spam , C’(x’) = legit

“Single-shot” analysis

A’(x) and it’s parameters

Do best response dynamics converge to a NE?

Attacker's Utility

- Attacker changes x to x'
  - Incurs a cost $c(x,x') = \sum_i c_i(x_i, x_i')$ for making modifications
  - Why do we need to account for the attacker's costs?
  - Example: #words added, #words modified etc.

- Receives a utility $U_A(y_C, y) \in \{-1,0,1\}$ where
  - y_C: predicted class by classifier *C*
  - y : true class

Implies that attacker only modifies spam emails

$U_A(y_C = legit, y = legit)$ =0

$U_A(y_C = legit, y = spam)$ =1

$U_A(y_C = spam, y = legit)$ =0

$U_A(y_C = spam, y = spam)$ =0

Attacker's Strategy

- Assume a spam message x classified by *C* as spam. Then:

$$\frac{P\{spam \mid x\}}{P\{legit \mid x\}} = \frac{P\{x \mid spam\} * P\{spam\}}{P\{x \mid legit\} * P\{legit\}} > 1$$

Assume=0

$$\Rightarrow \log\left(\frac{P\{spam \mid x\}}{P\{legit \mid x\}}\right) = \log\left(\frac{P\{x \mid spam\}}{P\{x \mid legit\}}\right) + \cancel{\log\left(\frac{P\{spam\}}{P\{legit\}}\right)} > 0$$

$$\Rightarrow \log\left(\frac{P\{x \mid spam\}}{P\{x \mid legit\}}\right) = \sum_i \underbrace{\log\left(\frac{P\{x_i \mid spam\}}{P\{x_i \mid legit\}}\right)}_{LO(x_i)} > 0$$

Attacker's Strategy

$$\Rightarrow \log(\frac{P\{x \mid spam\}}{P\{x \mid legit\}}) = \sum_i \underbrace{\log(\frac{P\{x_i \mid spam\}}{P\{x_i \mid legit\}})}_{LO(x_i)} > 0$$

- The attacker wants to change x to x' such that x' is classified as legit

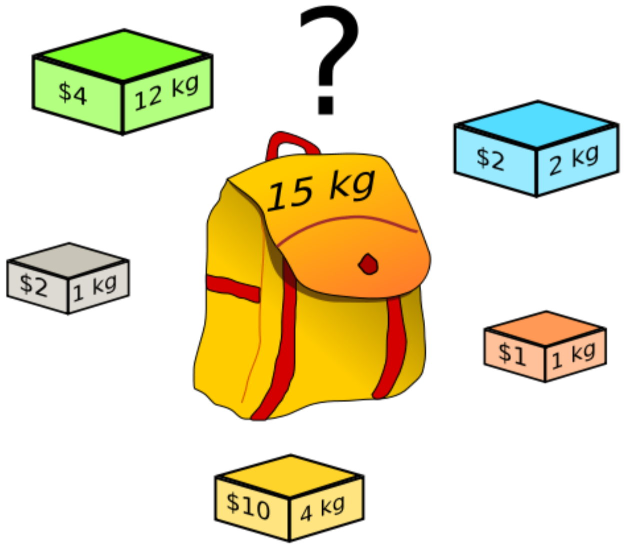
$$\sum_i \log(\frac{P\{x_i' \mid spam\}}{P\{x_i' \mid legit\}}) = \sum_i LO(x_i') < 0$$

$$\Delta = \sum_i LO(x_i) \qquad \delta_i = \max(LO(x_i) - LO(1 - x_i), 0)$$

Desired total reduction in sum-LO

Reduction obtained by adding/removing term *i*

Reduction to Knapsack Problem



- Knapsack of maximum weight 15 Kg
- N items, each of have a weight and reward
- Which items to put in knapsack such that:
  1. Weight of bag is less than 15 Kgs
  2. Reward is maximized

Attacker's Problem

- N terms, each term has "weight" δ_i and "cost" c_i if it is added/removed
- We need to add/remove enough terms such that the net weight exceeds
- While minimizing cost

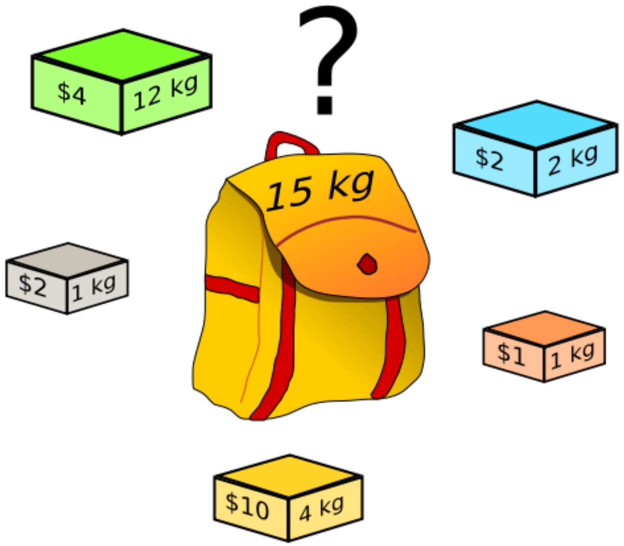
What if all terms cost the same?

Reduction to Knapsack Problem

Attacker's Problem

- N terms, each term has "weight" δ_i and "cost" c_i if it is added/removed
- We need to add/remove enough terms such that the net weight exceeds
- While minimizing cost

What if all terms cost the same?



How Does the Classifier Respond?

Classifier computes

$$\frac{P_A\{spam \mid x'\}}{P_A\{legit \mid x'\}} = \frac{P_A\{x' \mid spam\} * P_A\{spam\}}{P_A\{x' \mid legit\} * P_A\{legit\}}$$

Since the adversary only modifies the feature vector for spam emails, $P_A\{x' \mid spam\}$ is the only term that changes

$$P_A\{x' \mid spam\} = \sum_{x} P_A\{x' \mid x, spam\} P\{x \mid spam\} = \sum_{x:x'=A(x)} P\{x \mid spam\}$$

Sum over all possible feature vectors

Sum over all x which when modified by adversary yield x'

Relative cost of classifying legit emails as spam (FPs)

Results

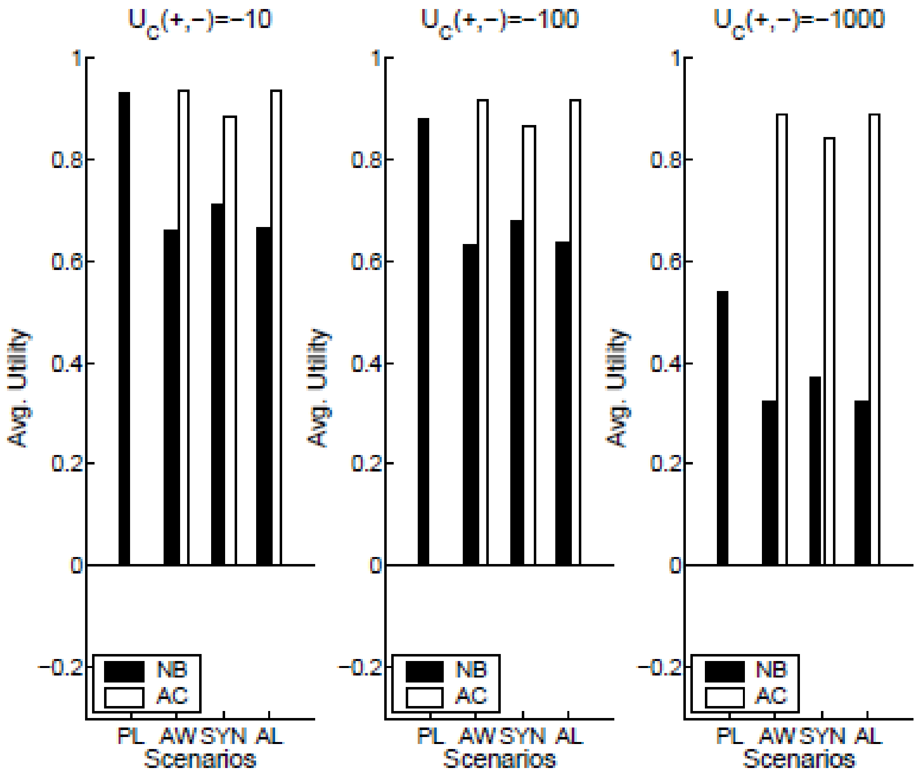


Figure 1: Utility results on the Ling-Spam dataset for different values of $U_C(+,-)$.

ADD WORDS (AW): Unit cost per word
ADD LENGTH (AL): Cost prop. To word length
SYNONYM (SYN): Unit cost per word

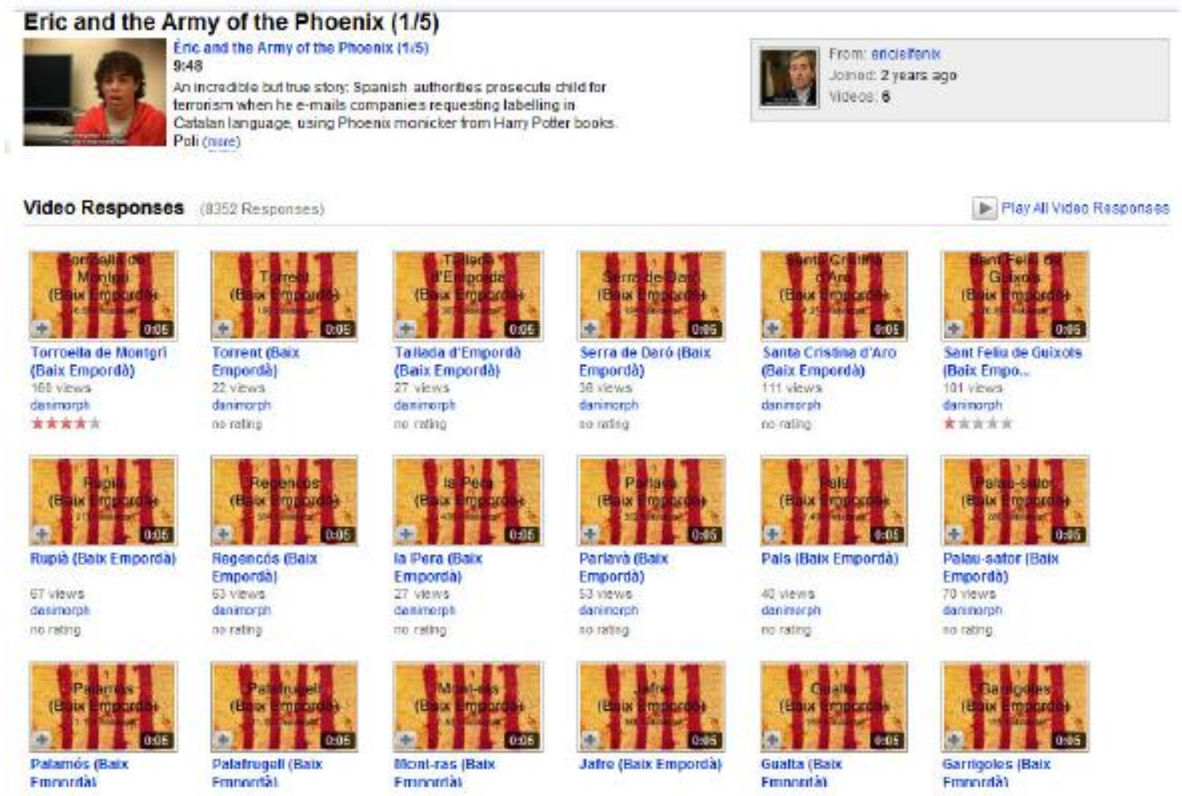
$U_C(+,-)$	10		100		1000	
Classifier	FN	FP	FN	FP	FN	FP
NB-PLAIN	94	2	124	1	165	1
NB-AW	481	2	481	1	481	1
AC-AW	93	0	123	0	164	0
NB-AL	477	2	477	1	477	1
AC-AL	94	0	124	0	165	0
NB-SYN	408	2	413	1	414	1
AC-SYN	164	1	196	0	229	0

Table 3: False positives and false negatives for naive Bayes and the adversary-aware classifier on the Ling-Spam dataset. The total number of positives in this dataset is 481, and the total number of negatives is 2412.

NB: Naïve Bayes, AC: Adversarially Trained

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?

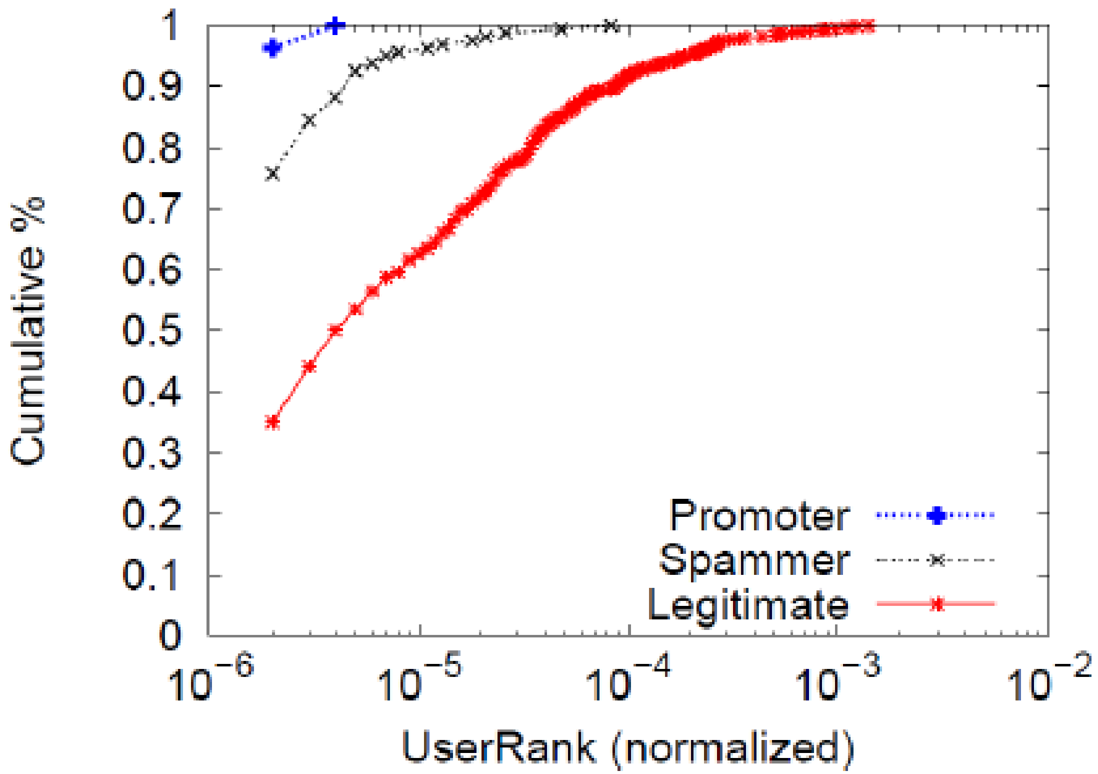
Features/Attributes Used

- 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

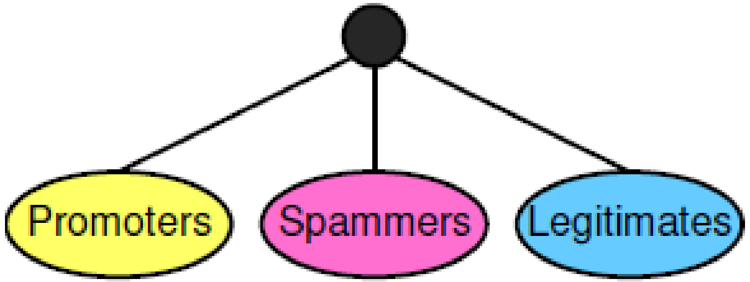
Attribute Set	Top 10	Top 20	Top 30	Top 40	Top 50
Video	9	18	25	30	36
User	1	2	4	7	9
SN	0	0	1	3	5

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

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