On Attacking Statistical Spam Filters

Gregory L. Wittel and S. Felix Wu
Department of Computer Science
University of California, Davis
One Shields Avenue, Davis, CA 95616 USA

Paper review by Deepak Chinavle

What is Spam

- unsolicited commercial email from someone without a pre-existing business relationship or, in practice, whatever email you don’t want
- Types
  - unsolicited mail containing adult content
  - unsolicited financial offers, etc.
  - unsolicited political or religious mail
  - consider unsolicited commercial email to be spam, even if it came from a sender with whom they’ve “already done business

Bayesian Classifier for Spam Detection (not part of the paper)

- Given – training set of messages with label as spam or no-spam.
- Message is broken into tokens
- Each token's probability given a class is calculated from from it's frequency in that class from the training data set.
- From Bayes theorem, we calculate the class having higher probability.
- Class with higher probability is considered to be the class of the message.

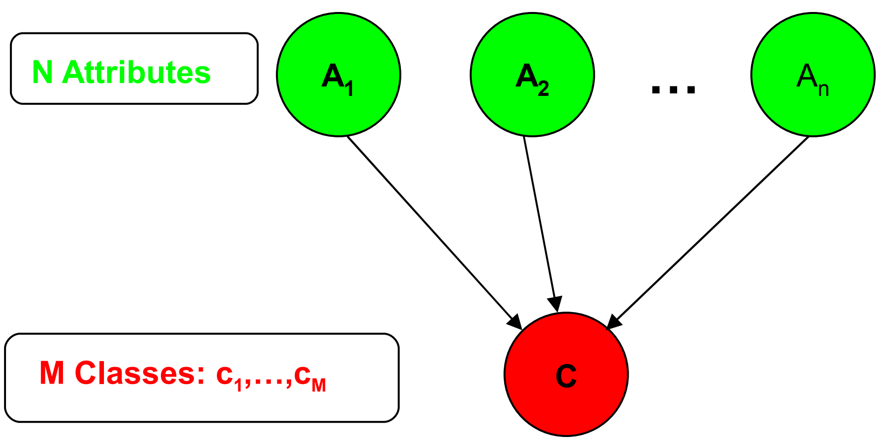
The probabilistic model of Naive Bayes classifier (not part of the paper)

- Assume attributes are independent for a given class
  - A **naïve** assumption

$P(c_k \mid a_1, a_2, \dots, a_n) = ?$ **for every k**

$$P(c_k \mid a_1, a_2, \dots, a_n) = \frac{P(c_k) * \boxed{P(a_1, a_2, \dots, a_n \mid c_k)}}{P(a_1, a_2, \dots, a_n)}$$

$$\boxed{P(a_1, a_2, \dots, a_n \mid c_k) = P(a_1 \mid c_k) * P(a_2 \mid c_k) * \dots * P(a_n \mid c_k)}$$



$P(C = c_1), \dots, P(C = c_k)$: known

Each $P(a_i \mid c_k)$ is easily to compute based on counts

Document classification using Naïve Bayes classifier (not part of the paper)

- The word "discount" appears really often in spam emails (i.e. 99%), but seldom see it in non-spam email (i.e. 1%)

- We need to train the *spam filter* these probabilities
  - We manually indicate whether a new email is spam or not → training set
  - P(discount | spam) = 0.99 , P(University | spam) = 0.05 , …

- After training, the *spam filter* has P(word | spam), for every word

- Filtering a new email consisting of *n* words

$$P(spam \mid words) = \frac{P(words \mid spam) * P(spam)}{P(words)} = \frac{\boxed{P(word_1 \mid spam) * \ldots * P(word_n \mid spam)} * P(spam)}{P(words)}$$

For example "If P(spam | words) > 0.95" → "Junk folder"

Ref: Wikipedia

Content-based Spam Filtering (Quick example of basic attack)

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

Feature Weights

2. cheap = 1.0
mortgage = 1.5

3. Total score = 2.5 > 1.0 (threshold)

Spam

Good Word Attacks (quick example of basic attack)

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

Corvallis OSU

Feature Weights

2. cheap = 1.0
mortgage = 1.5
Corvallis = -1.0
OSU = -1.0

3. Total score = 0.5 < 1.0 (threshold)

OK

Breaking filters – Attack types

- Tokenization :- Exploit the way features are extracted from messages ex. Putting extra spaces in words.
- Obfuscation :- obscure message contents using encoding ex. URL encoding.
- Weak statistical :- good word attacks with random words.
- Strong statistical :- good word attacks with educated guess of good words to use.

Breaking filters – Challenges faced by both the players

- Developers
  - Need to build accurate filters and at the same time consider possibility of attacks
  - Which training set to use ? This turns out to be very crucial in long run of the filter.
  - What if the training set used itself is erroneous ? This calls for trouble.
  - Lot of spam collections available publicly, what about ham collections ?
  - System wide Vs end user wise configuration of the filter.

Breaking filters – Challenges faced by both the players

- Spammers
  - Need to preserve the original message while still trying to make it look ham.
  - maximize audience size while keeping effort minimal.
  - Key is to find out the filter configuration or to find vulnerabilities in the training set of the filter.

Attack Methodology

- Simple attacks won't work, need to design advanced attacks.
- Designing repeatable attacks (which are missed and not caught subsequently)
- Getting feedback Ex. Yahoo mail.

Example Attack

- Spam example

  1 From: Kelsey Stone <bouhooh@entitlement.com>

  2 Subject: Erase hidden Spies or Trojan Horses from your computer

  3 Erase E-Spyware from your computer

  4 http://boozofoof.spywiper.biz

- To transform this spam, following two methods are used
  - Dictionary attack (random word from dictionary)
  - Common word attack (random words again but which are chosen from list of good words)

- The attacks were carried out against two types of filters
  - SpamBayes version 1.0a9
  - CRM114 release 20040312

- Data: 3000 spams from SpamArchive.org and 3000 hams from SpamAssassin

- n random words were added (n = 10, 25, 50, 100, 200, 300, 400) and for each n experiment repeated 1000 times

Results - SpamBayes

Words	Spam	Ham	Unsure
10	999 / 1000	0 / 0	1 / 0
25	937 / 772	0 / 0	63 / 228
50	484 / 16	0 / 0	516 / 984
100	22 / 0	0 / 943	978 / 57
200	0 / 0	269 / 1000	731 / 0
300	0 / 0	829 / 1000	171 / 0
400	0 / 0	858 / 1000	142 / 0

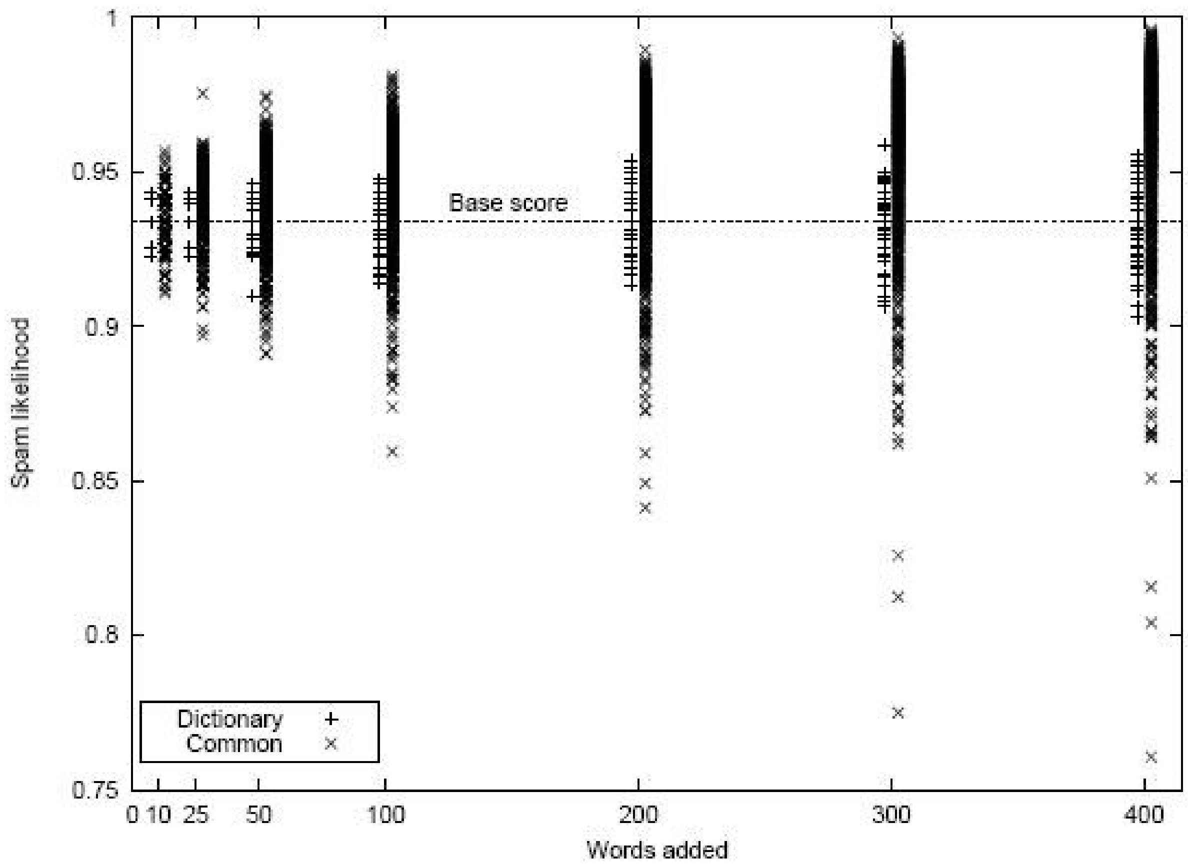
(a) Before training.

Words	Spam	Ham	Unsure
10	1000 / 1000	0 / 0	0 / 0
25	967 / 872	0 / 0	33 / 125
50	601 / 56	0 / 0	399 / 944
100	37 / 0	0 / 735	963 / 265
200	0 / 0	78 / 1000	922 / 0
300	0 / 0	625 / 1000	375 / 0
400	0 / 0	695 / 1000	305 / 0

(b) After training on source picospam.

More susceptible to good word attacks and when number of words added becomes more then 100.

Results - CRM114



- More robust to dictionary and good word attacks.
- Dictionary word attacks more confined (less spread out) compared to common work attacks.
- Bulk of common word attacks shift towards high spam score may be because of sequences of “bad” words.

Conclusion

- Most attacks still don't attack the statistical nature of the filter.
- Spammers modify spam only if they gain more then the effort required.
- Even the good word attacks failed with one of the filters tested.
- Effect of retraining with against attack messaged need to be studied.
- Need to look beyond statistical attacks like application of natural language processing to spam.
- Spam is a never ending game between spammers and anti-spammers.

Some comments

- Cross fold validation not done before giving the results (or at least not mentioned in the paper). Usually results are given after 10- fold cross validation.
- Only two filters used for testing, would have been interesting if filters using different algorithms for document classification were used like naïve bayes, support vector machines, maximum entropy etc.
- Spam and ham emails used were taken only form one source. By taking test data from various sources better models the real world situation.
- The results will be more interesting if the proportion of spam and ham emails is varied for training. For example, instead of using 50-50% ratio, use something like 60-40%. This tells us the bias used by the filter.