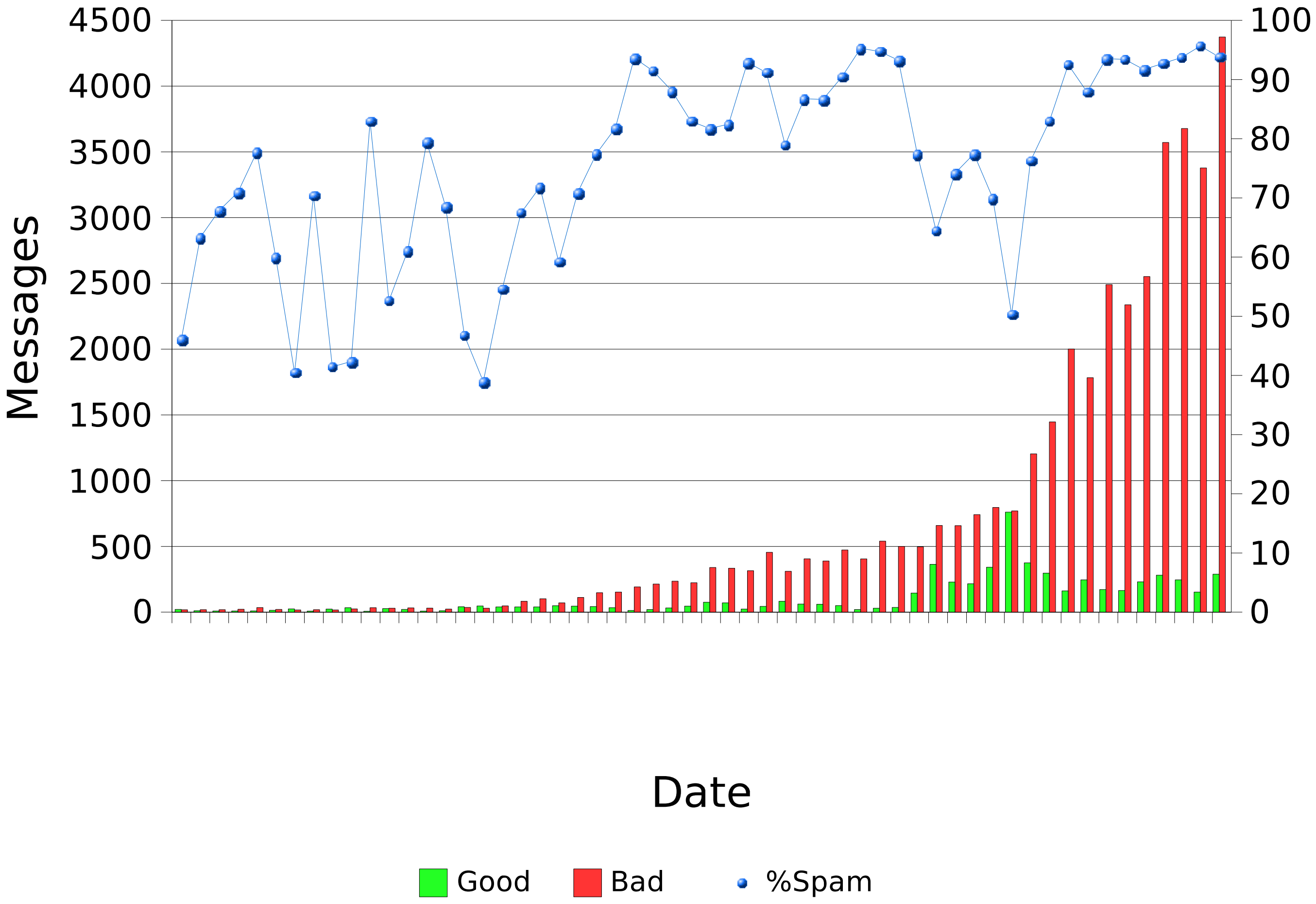
The Colour of Spam

Bimodal Classification of Electronic Mail using Statistical Bayesian Analysis

Donald Craig
Department of Computer Science
February 24, 2004

The Problem



- The volume of spam is increasing exponentially.
- 95% of my e-mail is spam.
- I receive about 150 to 200 spam messages per day. In two years, it may be ~3000 spams per day.

A Proposed Solution

- In August 2002, Paul Graham advocated Bayesian filtering as an effective means of combatting the spam problem. He didn't actually discover the technique, but he did help popularize it.

- The technique he described was quite simple with a claimed accuracy rate in excess of 99%!

- The solution works in two phases

  - Learning: The filter examines a collection of good and bad messages and derives statistical probabilities for all words based upon the relative frequencies of each word in the two collection of messages.

  - Filtering: The filter uses these statistical probabilities during its analysis of incoming messages to classify them as spam or non-spam.

The "Learning" Phase

Given:

$$
\begin{aligned}
msg_b &= \text{total number of bad messages,} \\
msg_g &= \text{total number of good messages,} \\
w_b &= \text{occurrences of word } w \text{ in bad messages,} \\
w_g &= \text{occurrences of word } w \text{ in good messages,}
\end{aligned}
$$

we can assign a probability to each word w based upon the relative frequencies that w occurred in all the bad and good messages.

$$P(w) = \max(\min(\frac{r_b}{r_g + r_b}, 0.99), 0.01)$$

where:

$$r_b = \min(\frac{w_b}{msg_b}, 1.0) \text{ and } r_g = \min(\frac{2w_g}{msg_g}, 1.0)$$

The further away $P(w)$ is from 0.5 (neutral), the more useful the word w is in determining whether or not the message is spam.

The "Learning" Phase (example)

Consider the word "`orders`" in my e-mail:

$$\begin{array}{lclclcl} msg_b & = & 17{,}867 & \texttt{orders}_b & = & 4{,}588 \\ msg_g & = & 4{,}425 & \texttt{orders}_g & = & 31 \end{array}$$

$$r_b = \min(\frac{4,588}{17,867}, 1.0) \approx 0.25679$$

$$r_g = \min(\frac{2 \times 31}{4,425}, 1.0) \approx 0.01401$$

$$\begin{array}{lcl} P(\texttt{orders}) & = & \max(\min(\frac{.25679}{.01401+.25679}, 0.99), 0.01) \\ & \approx & 0.94826 \end{array}$$

Therefore, the word `orders` is somewhat incriminating (but not as incriminating as other words such as `Xanax`, `HGH` and `optout`, which all have values of 0.99)

For other people, the probabilities will be different. *e.g.* for a student doing collaborative research into HGH, the word `HGH` would be less incriminating because this word may occur frequently in both her good and bad e-mail.

The Filtering Phase

Once we have generated a database of words and their corresponding probabilities, we can start to filter incoming messages.

When we receive a message, we identify the n most *interesting* words, w_i (Graham's algorithm sets $n = 15$). These are the words whose probabilities are furthest away from neutral. We then apply the following formula:

$$p = \frac{prod_1}{prod_1 + prod_2}$$

where:

$$prod_1 = \prod_{i=1}^{n} P(w_i) \text{ and } prod_2 = \prod_{i=1}^{n} (1 - P(w_i))$$

The closer that p is to one, the more likely the message is spam.

The Filtering Phase (example)

```
From kkjvt@etypejaguar.com  Thu Feb 12 21:04:56 2004
Return-Path: <kkjvt@etypejaguar.com>
Received: from etypejaguar.com (ip503cc0c7.speed.planet.nl [80.60.192.199])
  by garfield.cs.mun.ca (8.12.9-20030918/8.12.9) with SMTP id i1D0Yq97030948
  for <donald@cs.mun.ca>; Thu, 12 Feb 2004 21:04:54 -0330
Message-Id: <200402130034.i1D0Yq97030948@garfield.cs.mun.ca>
From: Sara <kkjvt@etypejaguar.com>
To: <donald@cs.mun.ca>
Subject: Fwd: Hey
Date: Thu, 12 Feb 2004 16:31:10 -0800
Mime-Version: 1.0
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD>
<META http-equiv=Content-Type content="text/html; charset=iso-8859-1">
<META content="MSHTML 6.00.2800.1269" name=GENERATOR>
<STYLE></STYLE>
</HEAD>
<BODY bgColor=#ffffff>
<DIV><FONT face=Arial size=2><A
href="http://wwx.img.netscape.net.c.medibargain.com/ks/lu.html"><IMG
alt="Try it and see" hspace=0
src="http://wwy.pc.yahoo.com.c.drug-source.com/omd.jpg" align=baseline
border=0></A></FONT></DIV>
<DIV><FONT face=Arial size=2></FONT> </DIV>
<DIV><FONT face=Arial size=2><A
href="http://wwx.img.yahoo.net.a.medibargain.com/ks/lu.html">
If you want to get huge savings on your medicine costs go look at us.
You'll pay far less. Plus you will not even need a medical office
visit. We take care of it all.We assure you that you will be very
pleased with our service. We deliver by overnight mail direct to your home.
</A></FONT></DIV></BODY></HTML>

uzqeyy
```

The Filtering Phase (example — cont'd)

$$
\begin{aligned}
\mathrm{P}(\texttt{wwx}) &= 0.99 \\
\mathrm{P}(\texttt{netscape}) &= 0.9401587088 \\
\mathrm{P}(\texttt{direct}) &= 0.0702056372 \\
\mathrm{P}(\texttt{IMG}) &= 0.9225396775 \\
\mathrm{P}(\texttt{office}) &= 0.0901773031 \\
\mathrm{P}(\texttt{pleased}) &= 0.1184996052 \\
\mathrm{P}(\texttt{ffffff}) &= 0.8808768573 \\
\mathrm{P}(\texttt{costs}) &= 0.1370993777 \\
\mathrm{P}(\texttt{Hey}) &= 0.1512649938 \\
\mathrm{P}(\texttt{Sara}) &= 0.1583361817 \\
\mathrm{P}(\texttt{jpg}) &= 0.834153526 \\
\mathrm{P}(\texttt{care}) &= 0.1672692044 \\
\mathrm{P}(\texttt{far}) &= 0.1673094906 \\
\mathrm{P}(\texttt{[80}) &= 0.8228206453 \\
\mathrm{P}(\texttt{src}) &= 0.8198552631
\end{aligned}
$$

The resulting bayesian value is 0.759014. If we set our *spam threshold* to, say, 0.5, then this message would be classified as spam by the filter.

This message is an exception, since most messages have a bayesian value extremely close to one or zero.

Modifications to Graham's Algorithm

Some modest modifications were made to Graham's original algorithm:

- Include the [character as being a word character.

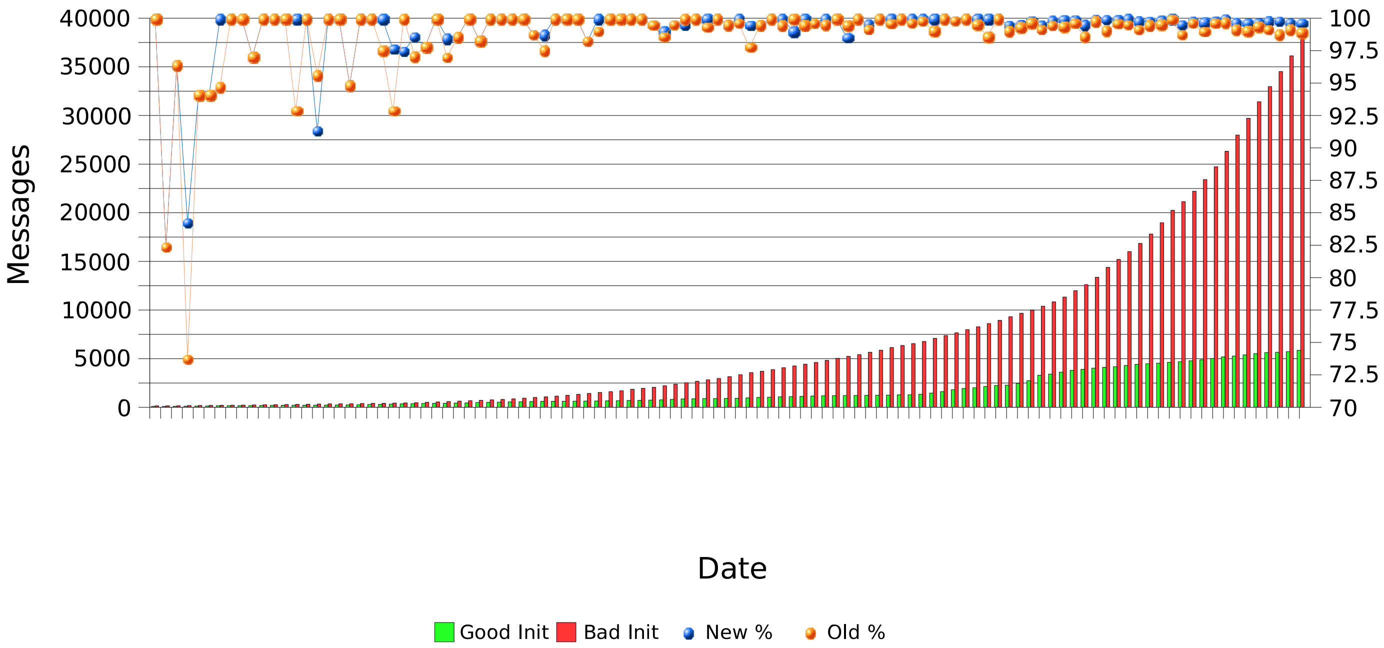
- Don't include the same word twice in the list of *interesting* words.

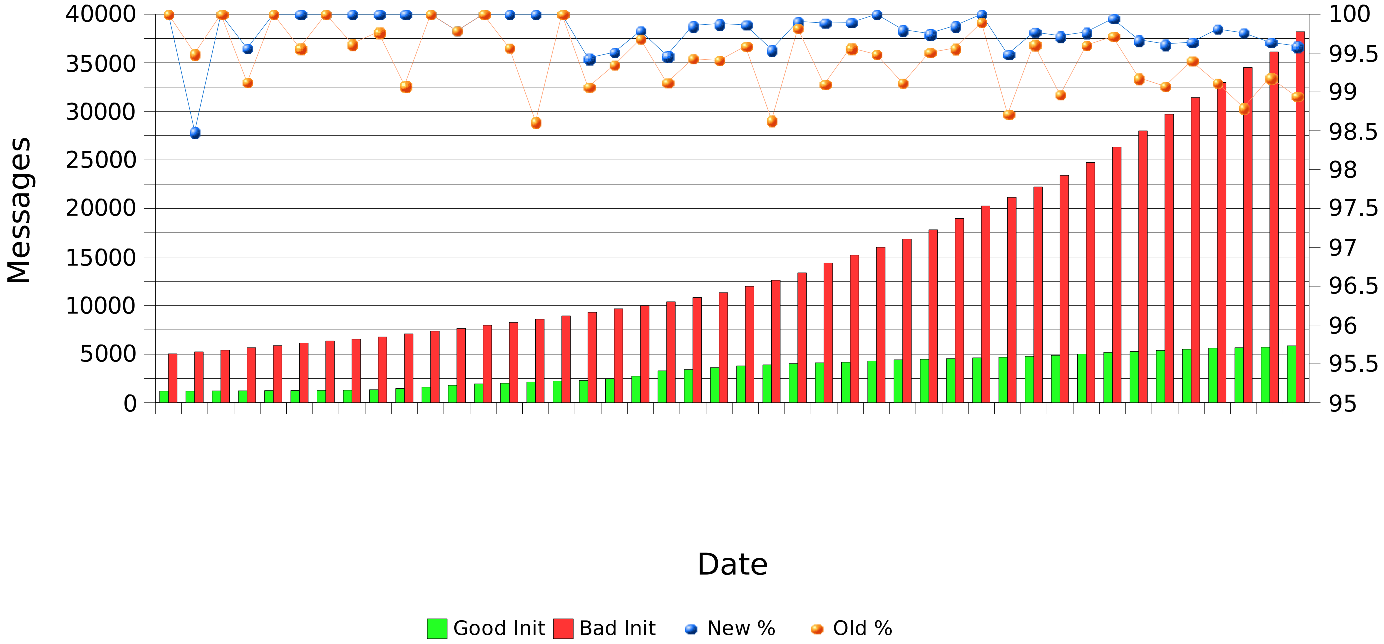
- To reduce false positives, for every extremely bad word that is used in the interesting word list, select two extremely good words.

- If Bayesian classification considers the message non-spam, then calculate a ratio of extremely bad words to extremely good words. If this ratio is high, then the message was probably poisoned. This helps reduce false negatives.

- My spam bayesian threshold level is currently set quite low (0.02). This has not resulted in any false positives (yet).

Performance





Conclusions

- Bayesian filtering can achieve a filtration rate in excess of 99%.
  - From August 2002 to Present : 99.507%
  - For 2004 : 99.712%
- Due to the highly personalized nature of the filter, bayesian filters can be very difficult for a spammer to defeat reliably.
- Problems
  - The "learning" phase can be somewhat computationally intensive. This can be mitigated by caching word counts for each mail folder.
  - The word probability file, which can grow fairly large (2MB), has to be processed on each inbound message.
- Spam is predominantly red.

References

- *A Plan for Spam*

  `http://www.paulgraham.com/spam.html`

- *buryspam.rb*

  `http://www.cs.mun.ca/~donald/buryspam/`

- *Mozilla Application Suite*

  `http://mozilla.org`

  Includes an e-mail client that uses bayesian filtering to identify spam automatically.

  Try it. It's free!