

CAN BAYESIAN FILTERS DELIVER?

Evolving from Content Filters to Bayesian Filters

The evolution of spam technologies quickly rendered simple, static content filters ineffective. The next generation of content-analysis engines developed for fighting spam, are of operate on the theory of probabilities: these engines look for indicative text, and set a spam-probability level based on those indications.

Bayesian filtering is one of the most advanced technologies in this class. It is a constantly trained and learning system based on users pointing out the engine's mistakes e.g. using a "This is spam" button for spam messages found in the inbox, and a "This is not spam" button for desirable email found in the junk folder.

Based on this information, the Bayesian engine builds a database of "good" and "bad" words. If a message consists of many spam-indicative words, and does not contain "non-spam" indicative words, it will be classified as spam.

What makes Bayesian filter better than content filters?

- It recognizes words that denote valid mail — to balance false positive mistakes
- It is dynamically updated (not based on static rules) — harder to fool than content filters
- It is sensitive to the user — spam for one person doesn't have to be spam for another

Bayesian Filtering — The Answer to Our Sorrows?

Learning, but not Self-Learning — Tutoring the Bayesian Means Hassles

Typically, Bayesian filters learn from their mistakes. This means you would first have to endure a low catch rate and many false positive mistakes, and only then will it gradually improve through your active participation:

- **The Learning curve's price tag:** A Bayesian system is not "out-of-the-box" ready. It takes thousands of messages to get the statistics to work for you (even according to the academic concept). Assuming 15-75 messages a day per average user — you are looking at months of low catch rate and too many false positives before you can rely on the filter.
- **Effectiveness depends on persistent user discipline:** it's not an easy task getting the "lazy" type of users (and most of us are, to be honest) to obediently press "spam" and "non-spam" over and over. Sadly enough, this effort is required on a consistent basis — not just in the initial "training period". Failing to do so not only makes the solution ineffective but will also severely increase the level of false positives.

Server-Side or Centralized Implementations; Losing the Bayesian Advantage

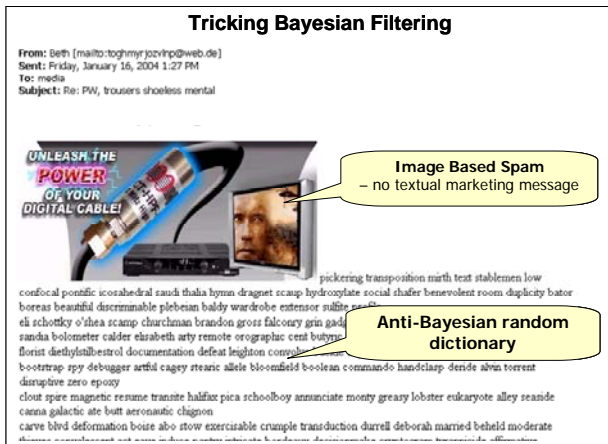
Understanding that the engine's training requirement is a serious issue, some vendors (Microsoft most noticeably) centralized the Bayesian update: instead of requiring users to define "spam" and "non-spam", a Bayesian-dictionary update is sent periodically (as a software update). However, this implementation neutralizes the core Bayesian advantages: it is not user-sensitive, not dynamic, and is easy for spammers to work around — as the same dictionary applies to all.

Others propose a 100% server-side Bayesian filtering implementation, and face major challenges:

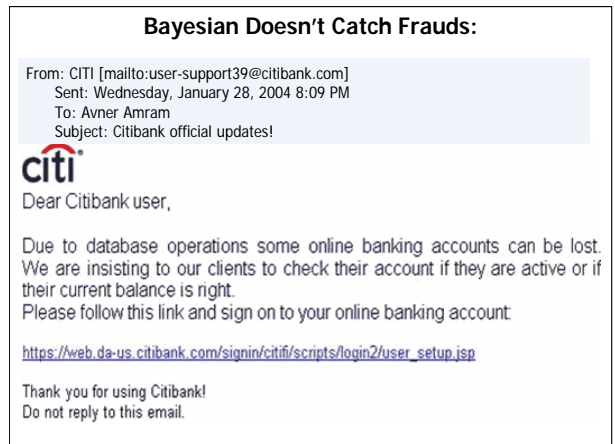
The best detection is on the day of installation: from then on, it constantly decreases: The Bayesian filter's effectiveness is strictly related to the end-users constant updating, according to personal experience. A Bayesian filter might be 'trained' in advance, but sooner rather than later it becomes out-of-date.

- **IT Effort Savvy:** instead of solving the training-effort issue, the problem is simply shifted from the users to the IT manager who has to constantly update the dictionary, and mark large quantities of "spam" and "non-spam".

- An inherent win-lose situation between users: Unlike viruses, spam is not equal for all. Server-side implementation of a Bayesian filter means an inherent win-lose situation between those users who consider a message spam and those who do not.



Example 1



Example 2

Spammers DO manage to work around Bayesian Filters

When originally developed, it was thought that working around Bayesian filters would be extremely difficult. However, with the coming release of Microsoft spam-filter (Bayesian-based), and other products spammers gone the extra mile rather successfully:

- Innocent looking spam:** In one example, Paul Graham (the Bayesian filter "guru") refers to a spammer trying to sell a vacation package. The message is written as if a friend is telling you about an incredible vacation from which he just returned. Another easy way is diluting the short spam marketing message by adding some 500 words from a recent CNN article at the bottom of the spam-message.
- Anti-Bayesian dictionary:** 700 randomly chosen words are inserted: no two emails produced this way are identical, making it impossible for a Bayesian engine to identify spam-indicative words in such a message (see example 1)
- Image based spam cannot be analyzed:** A Bayesian filter is 100% based on textual content. More and more spammers are basing their messages on non-textual file formats. In example 1 above, the marketing message is in the image, and the only textual content is completely innocent or random. Bayesian filters are totally defenseless against this growing trend.
- Ineffective against the most dangerous spam – fraud:** the nature of spam changes – whereas spam used to be 100% commercial, it is now becoming a common vehicle for fraud and phishing. These messages are of business nature, written in a business language; hence they do not contain trigger words that the Bayesian filter looks for. If the filter would have caught the fraudulent message in example-2 above (attempting to hijack the recipient's bank account information), it would have to filter other bank communications as well.

SUMMARY: NICE IDEA, DOESN'T DELIVER IN THE FIELD
 Spammers can work around it – they already do, more of them will in the future
 Training required: dependant on user discipline, drawn-out startup
 Problematic when implemented on the server side
 Cannot analyze image-based spam
 Ineffective against fraud

For more information, visit our website at: www.commtouch.com or contact us directly via email at: StopSpam@commtouch.com