WhyUseRules

Why use rules, instead of being a purely statistical filter?

SpamAssassin incorporates several means of detecting spam. Bayesian-style probabilistic classification is one of them, but not the only one.

SpamAssassin was actually one of the first open-source spam filters to include Bayesian-style probabilistic classification; we're not anti-Bayes in any way. However, we see Bayes as a **part** of the solution, and not something that fits all use cases.

Statistical filters require constant care

The way we see it is that a purely statistical filtering system is good if you have a techie user who's happy to spend time feeding it.

All of the Bayes-like statistical filters require constant training. This is fine if you're just filtering your own mail, and you enjoy tweaking your software setup – but if you're intending to use these for non-techie users, or users who don't enjoy spending time on filter training, then results will not be so good.

SpamAssassin's designed to give acceptable filtering without *any* training – and of course improve on that if you *do* spend the time to train it. We don't think the typical non-techie user should be required to train a filter by hand (unless they **want** to, of course).

Spammers are adapting

In addition, we expect that spammers are adapting, as they always have to each new form of filtering. Several spammer countermeasures have already been seen in use, aimed at Bayesian classifiers. SpamAssassin's 'wide-spectrum' approach is designed to catch these evasion techniques.

Swapping one form of load for another

A related criticism is that SpamAssassin is heavyweight, requiring more system load to classify mail than statistical filters do.

Our point of view is that spending more system load to get increased accuracy, without requiring user intervention, is an acceptable trade-off.

In addition, an often-overlooked aspect of statistical filter load is through disk space usage and disk I/O load. Some reports have indicated a company-wide statistical-filtering database can take up multiple gigabytes. With this amount of data, one must also consider the system and disk I/O load required to seek through it and return Bayes data in a timely fashion to the scanning software.

If the database seek time is too low, then using a statistical filter will simply exchange CPU and RAM load for I/O load and disk space; swapping one form of load for another.