Triumphs and Tribulations of Bayesianism in Spam Filtering

Scott Spanbauer

February, 2006

In 2002, software designer Paul Graham described a superior spam-blocking method based on the probability theorem of 18th-century mathematician Thomas Bayes. Based on Graham's thesis, other researchers subsequently produced a free spam-filtering program called Spambayes that, along with others like it, has proven to be an effective antidote to the current plague of unsolicited commercial e-mail. But although Spambayes demonstrates dramatically the predictive power of Bayes' theorem, its development process and that of similar spam filters have highlighted one of the biggest problems with the statistical method: implementation. Used intelligently, Bayes' equation can provide very accurate predictions about the truth of hypotheses, given certain evidence. But it can also fail miserably if that evidence is not properly encoded before input, and the equation's results are interpreted incorrectly.

Bayes' Theorem looks at what happened in a particular situation in the past (the evidence, E) and uses it to predict the probability of things going a particular way in that same situation in the future (the hypothesis, H). For example, the general probability of drawing the ace of spades from a fresh deck of cards is 1 in 52. Expressed as an equation, it looks like this: P(H) = 1/52, or .0192. As Curd and Cover explain, this is called the prior probability of our hypothesis, namely, that the next card we will draw from a fresh deck of cards will be the ace of spades (627).

Probability values lie between 0 and 1; the closer to 1, the more likely the truth of the hypothesis. If we now turn over 26 cards, none of which turns out to be the ace of spades, that evidence forces us to abandon our prior probability. Our likelihood of drawing the ace of spades from the remaining cards in the deck, given the evidence, is now 1 in only 26. This posterior probability of H, given E, expressed as

an equation, is P(H|E)=1/26, or .0385. The more our evidence increases the probability of H, the more it confirms our (still unlikely) hypothesis—that the next card we draw will be an ace of spades (627-628). If we manage to turn over 50 cards without getting the ace of spades P(H|E) climbs to .50. With only one card remaining it jumps to 1.0.

The first version of Bayes' Theorem states that P(H|E)=(P(E|H)*P(H))/P(E) (633). For our card example, with 26 cards turned over, that translates to P(H|E)=((26/26)*(1/52))/(26/51), or .0385. P(E), the probability of our evidence (26 cards), is 26 out of 52 cards. The probability of the evidence given our hypothesis (P(E|H)) is also pretty clear: if the card we're about to draw is definitely the ace of spades, then the probability that the last 26 cards drawn are not the ace of spades is 1. Curd and Cover confirm this (substituting T for H in their example): "...if T is a deterministic theory that deductively entails E, then everyone agrees that the correct value to be assigned to P(E|T) is 1" (634).

Unfortunately, this is where the deck of cards ceases to be an interesting case study of Bayesianism. A deck of cards hands us a pile of evidence up front—we know it contains 52 cards arranged in four suits, numbered in thus and such a way. We don't need a complex equation to predict the likelihood of drawing a particular card. Spam-filtering is a much more difficult problem, however, where evidence is abundant, but often seemingly inconclusive and even contradictory. Messages containing the word Viagra are almost certainly spam--unless they happen to contain the text of this paper. At the same time, both spam and non-spam messages contain many ordinary and unusual words in similar ratios. The only thing we can (fortunately) say for sure about them is that we know spam when we see it: a message is either spam, or it is not.

Human beings who sort through their inboxes use a number of cues to separate the spam from the non-spam, including the words in the message body, the subject line, and even the message formatting. How do we know that a message containing the word Viagra in the subject line is likely to be spam?

The answer is that we have seen so many messages that turn out to be pitches for Viagra, and not messages from an actual person we know, that we judge any new messages containing the word Viagra to be spam. If we were to analyze hundreds of our incoming messages, comparing the number containing the word Viagra to the number of those messages subsequently confirmed by us to be spam, we could determine the probability of future messages that mention Viagra being spam, a number that would probably be very close to 1.

We could then create a rule that deletes all messages containing Viagra—akin to the way the first generation of list-based anti-spam programs worked. Unfortunately, those programs weren't terribly accurate, forcing their users not only to continue to slog through undetected spam, but to also to check for important messages marked mistakenly as spam, false positives. Here, Bayes' Theorem provides a superior filtering method. By analyzing every word in every message, then asking the human e-mail recipient to confirm whether those words came from spam or non-spam messages, programs like Spambayes quickly learn to separate good mail from bad with a high degree of accuracy. Most importantly, they have a low rate of false positives, which are "an order of magnitude worse than receiving spam," according to Paul Graham (2002). He also notes:

> Because it is measuring probabilities, the Bayesian approach considers all the evidence in the email, both good and bad. Words that occur disproportionately *rarely* in spam (like "though" or "tonight" or "apparently") contribute as much to decreasing the probability as bad words like "unsubscribe" and "opt-in" do to increasing it. So an otherwise innocent email that happens to include the word "sex" is not going to get tagged as spam (2002).

Shortly after publishing "A Plan for Spam," Graham was surprised to learn that two other teams of researchers had already attempted Bayesian spam filtering, with mediocre results. Compared with Graham's own initial Bayesian test filters, which trapped 99.5% of spam with only .03% false positives,

the better of the two teams' filters caught only 92% of spam messages, with 1.16% false positives, a difference big enough to make the first worth using, and the second worse than not filtering at all. When Graham investigated the better-performing team's methods, he found several likely sources of error stemming from unfortunate design decisions that in two cases resulted in the filter throwing out data that might actually help identify spam. The team also failed to train its filter with a sufficiently large collection of e-mail messages, according to Graham (2003).

But Graham's Bayesian algorithm, at least in its initial form, suffered from flaws of its own. As two of Spambayes' developers, T.A. Meyer and B. Whately state in a 2004 paper presented to the Conference on E-Mail and Anti-Spam, Graham's initial theory had a tendency to produce scores of either one (where the message is identified as spam) or zero (non-spam), with few messages falling in the middle. “As a result,” they note, “when the system was wrong, it was completely confident in its (incorrect) score” (1). Spambayes' developers added chi-square analysis of results to better identify messages that don't clearly fall into a spam or non-spam category, with a resulting false-positive rate of 0% (3).

The developers continue to improve Spambayes by fine-tuning how it identifies bits of text prior to analysis, and how statistical analysis is used to minimize false positives. But one area where no one, including Paul Graham, appears to be making improvements, is in Bayes' theorem itself. Despite its 18th -century roots, the equation has shown itself to be a match for one of the most difficult modern challenges.

Works Cited

Cover, J and M. Curd (1998) "Bayesianism Commentary," in Philosophy of Science: the Central Issues, Curd and Cover eds. W. W. Norton & Company. pp. 627-643.

Graham, Paul, (August, 2002) “A Plan for Spam.” <http://www.paulgraham.com/spam.html>.

Graham, Paul, (January, 2003) “Better Bayesian Filtering” <http://www.paulgraham.com/better.html>.

Meyer, T.A., and Whateley, B., (2004), “SpamBayes: Effective open-source, Bayesian based, email classification system.” Conference on Email and Anti-Spam, July 30 and 31, 2004. <http://ceas.cc/papers-2004/136.pdf>.