BAYES' THEOREM & BAYESIAN INFERENCE

An early example of algorithms that track authenticity is email spam filters, motivated by the huge spam problem starting in the '90s. In 2010, close to 90% of emails were spam, with annual cost to society of $20 billion (see References).

A filter tries to keep emails you want (eg, friends, ‘legitimate’ unsolicited emails) from those you don’t (eg, bulk ads). Clearly this is subjective — do you consider your friend’s bulk marketing email to be spam?

Let’s build a simple spam filter algorithm.

Given an incoming email: $S$ is the event "this email is spam". The other possibility is "not $S". The Bayesian approach tracks a probability of, or “numerical belief in,” $S$. This probability will change in the light of new input, just as your opinion of an email crystallizes as you read through it.

All probabilities lie between 0 and 1:

\[ 0 \rightarrow \text{certainly not spam} \rightarrow \text{pretty certain it's spam} \rightarrow 1 \rightarrow \text{certainly is spam}. \]

We’ll use $p(S)$ to mean prior probability of being spam, ie before examining the email. Given the above statistics, $p(S) = 0.9 = 90\%$ is a good estimate of this prior.
Now, let U be the event "this email contains the word 'urgent'".
We need to know how common 'urgent' is in spam & in non-spam.
To estimate this, say we collect 1000 random emails and find (say) the following:
\[
\begin{array}{c|cc}
& \text{not S} & S \\
\hline
\text{all emails} & 100 & 900 \\
\text{containing 'urgent'} & 10 & 360 \\
\end{array}
\]
From this "training data" we estimate \( p(U|S) = \frac{360}{900} = 0.4 \), i.e. 40% of spam contains 'urgent'.
\( \text{called conditional probability of U occurring given S.} \)
The other data we need is the probability of U without knowing anything about S.
We can also read this from our table: \( p(U) = \frac{10 + 360}{1000} = 0.37 \)
Now, say the incoming email contains 'urgent', the algorithm computes \( p(S|U) \) via Bayes' theorem:
\[
p(S|U) = \frac{p(U|S) \cdot p(S)}{p(U)} = \frac{0.4 \cdot 0.9}{0.37} \approx 0.973, \text{ i.e. 97.3}\% chance of being spam.}
If instead it doesn't contain 'urgent', again Bayes gives the posterior:
\[
p(S | \text{not } U) = \frac{p(\text{not } U | S) \cdot p(S)}{p(\text{not } U)} = \frac{0.6 \cdot 0.9}{0.63} \approx 0.857 \text{ high, but less than our prior.}
\]
The algorithm must pick a threshold; if posterior > 0.95 (say) it goes to spam.
So far, this is not a great filter: you miss every email with 'urgent' in it!
Its "false positive rate" is \( \frac{10}{100} = 0.1 = 10\% \), way too dangerous.
But we can do much better: Bayes lets you update the posterior given new data:
Let V be "the email contains 'viagra'" ← this hardly ever occurs in non-spam!
A wrong but useful model is to assume \( p(U \text{ and V} | \ldots) = p(U | \ldots) \cdot p(V | \ldots) \).
\( \text{this is called independence of U & V.} \)
Bayes then gives, for an email with both 'urgent' & 'viagra',
\[
p(S | V \text{ and } U) = \frac{P(V \text{ and } U | S)}{P(V \text{ and } U)} p(S) = \frac{P(V | S) \cdot P(U | S)}{P(V) \cdot P(U)} p(S)
\]
"update" to the posterior given our old posterior, the new input V.

By continuing this way with a pool of many words common in spam, one gets quite a reliable Bayesian spam filter.

Notes:

- there can be many unintended consequences! Legitimate emails can end up in spam, but also spammers change tactics by misspelling words (Viagra) or including random text ("Bayesian poisoning") — it's an arms race, with evolving viral warfare.

- real filters are fancier, updating the posterior using phrases, URLs, blacklisted senders, presence of CAPS, etc...

- Here's the derivation of Bayes' theorem; it is nothing more than the rules of probability. We equate the two ways of factoring the joint probability,

\[
p(U \text{ and } S) = p(U | S) p(S)
\]

\[
p(U \text{ and } S) = p(S | U) p(U)
\]

equating these two, and rearranging gives Bayes' theorem, as stated on previous page.

This shows the two orderings of events to get to "U and S".