Spam Filter Math

Andrew Candela

August 31, 2016

The Idea

Assuming that I have a vocabulary of words $\mathcal{W}$, the probability of a message containing some vector of words given that the message is spam (and similarly for a message that is not spam) is

$$P(\, W_1 = x_1,\, \ldots\, , W_n = x_n \mid S\,) = \prod_{w_i \in \mathcal{W}} P(\, W_i = x_i \mid S\,)$$

assuming that each of the W_i's are independent [1]. Here each W_i is a RV where

$$W_i = \begin{cases} 1, & \text{if } w_i \text{ is in the message} \\ 0, & \text{otherwise} \end{cases}$$

Ultimately what we want is to be able to assign a probability that a message is spam given a vector of words. We want to estimate $P(\, S \mid \boldsymbol{W} = \boldsymbol{x}\,)$. Remember that we can use Bayes theorem to write:

$$\begin{aligned} P(\, S \mid \boldsymbol{W} = \boldsymbol{x}\,) &= \frac{P(\, \boldsymbol{W} = \boldsymbol{x} \mid S\,)P(\, S\,)}{P(\, \boldsymbol{W} = \boldsymbol{x}\,)} \\ &= \frac{P(\, \boldsymbol{W} = \boldsymbol{x} \mid S\,)P(\, S\,)}{P(\, \boldsymbol{W} = \boldsymbol{x} \mid S\,)P(\, S\,) + P(\, \boldsymbol{W} = \boldsymbol{x} \mid \neg S\,)P(\, S\,)} \\ &= \frac{P(\, \boldsymbol{W} = \boldsymbol{x} \mid S\,)}{P(\, \boldsymbol{W} = \boldsymbol{x} \mid S\,) + P(\, \boldsymbol{W} = \boldsymbol{x} \mid \neg S\,)} \end{aligned}$$

So we can re-write the probability that a given message composed of words $w \in \mathcal{W}$ is spam as:

$$P(\, S \mid \boldsymbol{W} = \boldsymbol{x}\,) = \frac{\prod_{w_i \in \mathcal{W}} P(\, W_i = x_i \mid S\,)}{\prod_{w_i \in \mathcal{W}} P(\, W_i = x_i \mid S\,) + \prod_{w_i \in \mathcal{W}} P(\, W_i = x_i \mid \neg S\,)} \tag{1}$$

Assuming we have a training set of emails $\mathcal{E}$ composed of spam messages $\mathcal{S}$ and real messages $\mathcal{R}$. Let's estimate the probability of seeing spam word w_i in a spam message $s \in \mathcal{S}$ as:

$$P(\, W_i = 1 \mid S\,) = \frac{\text{number of spam emails that contain } w_i}{\text{total number of spam emails}}$$

or, in fancy terminology:

$$P(\, W_i = 1 \mid S\,) = \frac{|\{s \in \mathcal{S} : w_i \in s\}|}{|\{\mathcal{S}\}|} \tag{2}$$

[1]The assumption of independence of the W_is is a simplifying one, but perhaps not a realistic one. I suppose I could try to cluster words into 'phrases' and then treat each phrase as an event. I would be able to apply this model and would address the concern that my independence assumption is invalid. Let's first get this to work for single words, and then move on from there.

Practical Concerns

Underflow

In order to determine $\prod_{w_i \in \mathcal{W}} P(W_i = x_i \mid S)$ and $\prod_{w_i \in \mathcal{W}} P(W_i = x_i \mid \neg S)$ we will be multiplying a bunch of floating point numbers together. This may cause a problem, so let's do the following. Let's denote ρ_i to be our estimate that the ith word is in a spam message. In order to compute the product of the ρ_i's, we can use:

$$\prod_i \rho_i = exp\left[\sum_i \ln(\rho_i)\right] \tag{3}$$

Smoothing

Imagine your message contains a word w_k that is not in the training set. This is problematic in two ways. If that word is not present in the training set, then our estimate $|\{s \in \mathcal{S} : w_k \in s\}|$ of $P(W_k = 1 \mid S)$ would be 0. We would similarly estimate $P(W_k = 1 \mid \neg S)$ to be 0. This will cause a division by zero error in equation (1). If $w_k \notin \mathcal{S}$ but $w_k \in \mathcal{R}$ then we will assign zero probability that this message is spam. Ideally, we would want to admit the possibility that this message is spam, and assign a small but nonzero probability to it.

We accomplish this by adding a 'psuedocount' k. Let's now define our probabilities as:

$$P(W_i = 1 \mid S) = \frac{|\{s \in \mathcal{S} : w_i \in s\}| + k}{|\{\mathcal{S}\}| + 2k} \tag{4}$$

Conclusion

Looks like we have everything we need to start evaluating our emails. Let's look for a training set and get started searching for spam!