CS2013 Tutorial on applications of Bayesian Reasoning **+ Answers**

This tutorial applies Bayesian Reasoning to spam filtering, an important practical problem in computing. The tutorial is adapted from Ken Rosen, "Discrete Mathematics and its Applications", 6th edition, Chapter 6.3 (and exercises).

The purpose of a spam filter is to estimate the probability that a given message is unwanted, and to use this probability to decide whether to mark the message as spam. We assume here that the spam filter bases its decision on the words that occur in the body of the message. Some words might make it more likely that a message is spam (or not). Words like "special", "offer", and "friend" might be suspicious, for instance.

Suppose, for simplicity, we focus on one individual word w. We abbreviate the event that a given message m is spam as S, and the event that m contains w (at least once) as E. Then one problem at the heart of spam filtering is to estimate p(S|E). The second problem is to decide upon a wise threshold t such that when p(S|E)>t, you classify the message as spam. (If you choose t too high, you will overlook some spam messages; if you choose t too low, you will misclassify some messages as spam – which is probably worse.) In this tutorial, assume t=0.95.

The idea that we explore in this tutorial is to estimate p(S|E) using Bayes' Law. The reason is that p(S|E) itself might be quite difficult to estimate directly. (How do we find a large set of messages that contain w? Note that this is a different set for each w, so if you're looking at different words you would have to do a lot of work using this approach.) On the other hand, Bayes' Law enables us to compute p(S|E) from some other numbers that are a bit easier to obtain.

First a corpus is collected consisting of two parts. Part B ("Bad") is a set of messages that users have marked as spam; pat G ("Good") is a set of messages that users have marked as non-spam. This twin corpus is first studied informally to identify words that might give away whether a message is in G or B. The next step is to focus on a given word w and to count how often it occurs in B and G:

n(B,w) = the number of messages in B that contain w
n(G,w) = the number of messages in G that contain w

We estimate the probability p(E|S) that an incoming spam message contains w, to equal the proportion n(B,w) / |B| (vertical bars stand for "the number of elements of"). Likewise we can estimate the probability p(E|not-S) that an incoming non-spam message contains w as n(G,w) / |G|. To compute p(S|E), we need some information about p(S) as well. Below we discuss two scenarios: one if which we make a guess about p(S), and one in which we have actual information about p(S). Both make use of Bayes' Law, in the following shape:

$$P(S|E) = \frac{p(E|S) * p(S)}{(p(E|S) * p(S)) + (p(E \mid \text{not-}S) * p(\text{not-}S))}$$

Question 1: Derive this version of Bayes' Law from the more common version of this Law, which divides p(E|S) * p(S) by p(E).

Bayes' Law says

$$P(S|E) = \frac{p(E|S) * p(S)}{P(E)}$$

For any Boolean variable X, we have:
P(E) = (by the rule of Marginalization) =
P(E∩X) + P(E∩not-X) = (by the Product Rule) =
(P(E|X)*P(X)) + (P(E|not-X)*P(not-X)).
Substitute S for X in this formula and we obtain the denominator of the fraction in the question.

Question 2: Suppose you have no information at all about p(S). In this case, we could assume that a message is equally likely to be spam or not to be spam; in other words, assume p(S)=p(not-S). Use this information to simplify the equation above. (When you're done, neither the numerator nor the denominator should contain a product.)

Given p(S)=p(not-S), the equation above can be simplified immediately to

$$P(S|E) = \frac{P(E|S)}{P(E|S) + p(E|\text{not-}S)}$$

To see this, replace P(E|S) and P(E|not-S) in the original formula by the same constant. (If it helps, you can replace both P(S) and P(not-S) by 0.5)

Question 3: Suppose that the word "free" occurs in 2500 of the 20.000 messages in B (which are known to be spam), and in 50 of the 10.000 messages in G (which are known not to be spam). A message comes in that contains the word "free". Continuing to make the assumption that p(S)=p(not-S), compute p(S|E), and determine whether the message is classified as spam.

Using the formula obtained in Q2, using Free to say that the word "free" occurs in the message, then substituting the frequencies we find, we get

$$P(S|E) = \frac{P(\text{Free}|S)}{P(\text{Free}|S) + p(\text{Free}|\text{not-}S)} = \frac{0.125}{0.0005+0.125} = \text{approx } 0.962$$

0.962 > 0.95, so given this threshold t, the message is classified as spam.

Question 4: We now turn to the second scenario, where we have actual information about p(S). Suppose you have collected descriptive statistics over a large body of emails and found that spam is relatively rare: non-spam is 9 times as frequent as spam. With this information, answer the same questions as in question 3. Before you start calculating, do you expect p(S|E) to be greater or smaller than in question 3?

This time, P(S)=0.1 and P(not-S)=0.9. This time the original equation cannot be simplified as before. Using the original formula and substituting 0.9 for P(S) and 0.1 for P(not-S), we get

$$\mathbf{P(S|E) = \frac{p(E|S) * 0.1}{(p(E|S) * 0.1) + (p(E \mid \text{not-}S) * 0.9)} = \frac{0.125 * 0.1}{(0.125*0.1)+(0.005*0.9)}}$$

This equates to 0.0125/0.0170, which is approx 0.735. This time the prediction is not confident enough to reach the threshold for classifying the message as spam. – Another reminder of the importance of the prior probability (in this case: P(S)).

Question 5: In reality it would be unrealistic to expect any one word to be a good predictor of whether a message is likely to be spam. Let's look briefly at what happens when 2 words are taken into account. Suppose E1 and E2 are the events that an incoming message contains the words w1 and w2, respectively. For simplicity, assume that E1 and E2 are independent events, and that p(E1|S) and p(E2|S) are independent events as well. Finally (as in question 2), assume that p(S) = p(not-S). Now show in a number of steps that

$$P(S|E1 \cap E2) = \frac{p(E1|S) * p(E2|S)}{p(E1|S) * p(E2|S) + (p(E1|\text{not-}S) * p(E2|\text{not-}S)}$$

$$\mathbf{P(S|\ E1 \cap E2) = \frac{P(E1 \cap E2 \mid S) * P(S)}{P(E1 \cap E2)}}$$

$$\mathbf{= \frac{P(E1 \cap E2 \mid S) * 0.5}{P(E1 \cap E2|S)*0.5 + P(E1 \cap E2|\text{not-}S)*0.5}}$$

$$\mathbf{= \frac{P(E1 \cap E2 \mid S)}{P(E1 \cap E2|S) + P(E1 \cap E2|\text{not-}S)}}$$

$$\mathbf{= \text{(independence)} \frac{P(E1|S) * P(E2|S)}{(P(E1|S)*P(E2|S)+(P(E1|\text{not-}S)*P(E2|\text{not-}S)}}$$

Question 6 (optional): Can you think of further ways to improve the spam filter?

Calculating the analogous probabilities for conjunctions of more than 2 words

Expanding the sets of interesting words by adding synonyms (assuming synonyms affect the likelihood of being spam in similar ways), or looking at all words that contain the symbol "$" or "£", for instance.

Taking larger Good and Bad samples, to get a better idea of whether a given word is indicative of spamhood

Taking into account how often an interesting word occurs in a message (instead of only taking into account whether it occurs in the message at all).

Weighing words in important positions (e.g., subject line) more heavily

Etc.

END OF TUTORIAL