Gerald's Column
by Gerald Fitton

In this article I shall get down to the 'nitty gritty' of how SpamStamp works. Before I started writing I was sure that I could 'polish it off' with one article—but I found that I couldn't get the size down to something acceptable and still keep it 'interesting'. So, with some regret, I have had to restrict the amount of material included in this month's column.

I do hope that this series has not gone on too long. If you have comments or questions then please write to me even if you think the answers to your questions might appear later.

The 'Simplified' Bayes' Equation

After a lot of 'heavy' algebra last month we (almost) arrived at a very simple equation which, undoubtedly, is the best format for considering how SpamStamp works.
This version of Bayes' formula is:

$$P = \frac{1}{(1 + r)}$$

Simplified Bayes' Equation

Of course, in order to use it we need to know the meanings of P and r.

We have an email. We don't know if it is Spam or Ham ("Ham" is Jan–Jaap's label for those emails which are not Spam). We look at all the words in the email, we check how often they have appeared in both the Spam and Ham emails that we know about, we do a few sums to find the value of r, enter this value of r in the equation and then ... we find P.

P is the probability that this particular email is Spam. Put that way it's easy, isn't it?

Calculating r

Last month I (almost) defined r as $r = r_0 * r_1 * r_2 * r_3 * r_4 * r_5$ etc. Each value of r_n is calculated for each word in the email. These values are multiplied together (in any order) and the answer is the Ratio, r, which is used to find the Probability that the email is Spam.

$$r_n = \frac{P(E_n|H_2)}{P(E_n|H_1)}$$

Calculating r_n

In his original article Jan–Jaap used the word "feet" as his example. We shall do the same. In the formula which I have labelled "Calculating r_n" the Event, E_n, is the inclusion of the word "feet" in the email, Hypothesis H_1 is that the email is Spam and Hypothesis H_2 is that the email is Ham. We shall use the original values quoted by Jan–Jaap to find r_n.

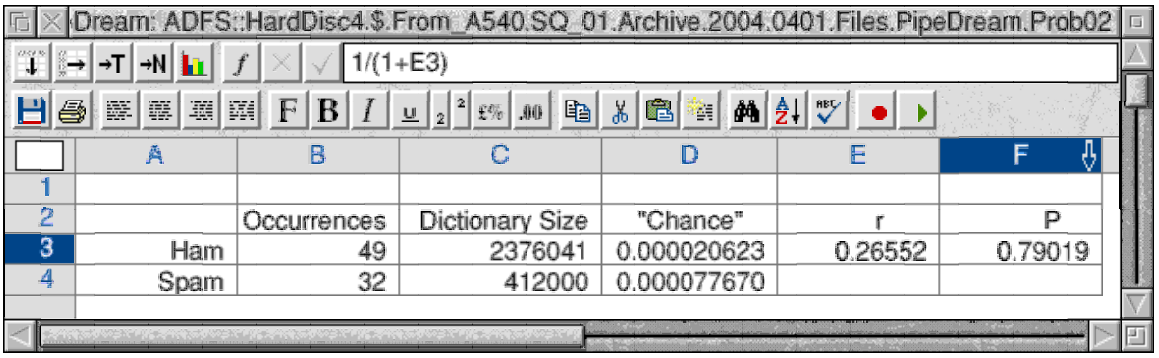
Jan–Jaap reported that, in his Spam database, the word “feet” had occurred 32 times and in his Ham database, 49 times. There were, at that time, 412 000 words in his Spam database and 2 376 041 words in his Ham database.

The initial calculations are:

$P(E_n|H_2)$ = 49/2 376 041 = 0.000 020 623 and
$P(E_n|H_1)$ = 32 / 412 000 = 0.000 077 670

Using these values r_n = (0.000 020 623)/(0.000 077 670) = 0.26552.

If this were the only value of r in the chain of ratios, then this value of r could be converted to a probability using what Jan–Jaap refers to as “Bayesian inversion”. The result of this “inversion” is that P = 1/(1 + r) = 0.790. This value, 0.790, is that quoted by Jan–Jaap as the “probability that the email is spam” based solely on the inclusion of the word “feet”.



	A	B	C	D	E	F
1						
2		Occurrences	Dictionary Size	"Chance"	r	P
3	Ham	49	2376041	0.000020623	0.26552	0.79019
4	Spam	32	412000	0.000077670		

The PipeDream spreadsheet, [Prob02], shown above can be used to find the values of both r and P when the basic data, occurrences and dictionary size, are available. I have illustrated its use with Jan–Jaap’s values for the word “feet”. You can enter your four numbers in the block B3C4 and PipeDream will calculate “Chance” (column D), r and P.

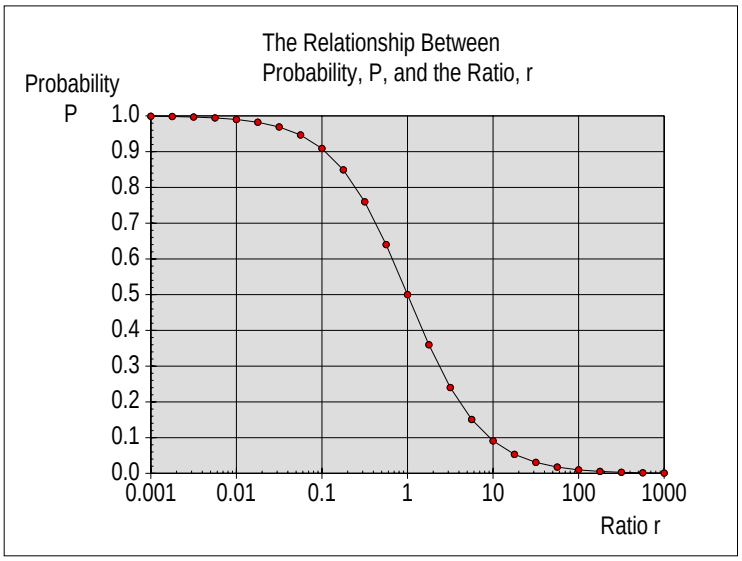
Values of r and P

Look again at the formula I have called “Simplified Bayes’ Equation”.
The Ratio, r, can take values between zero and infinity.
The Probability, P, can take values between 0 and 1.

When r = zero, the probability, P = 1, certainty. The email is certainly Spam.
When r = infinity, the probability, P = 0, impossibility. The email is Ham.
When r = 1 the probability, P = 0.5. There is a 50:50 chance the email is Spam.

The Relationship Between P and r

I created the spreadsheet, [Prob01], using PipeDream and saved the graph labelled “The Relationship Between Probability, P, and the Ratio, r” as a drawfile from PipeDream. Note the logarithmic scale for the ratio, r; it makes the graph symmetrical. Later we shall discover why this is an interesting and useful feature of the relationship between r and P.



As you can see from the graph, what I might call the ‘uncertain’ values of r lie between 0.01 and 100. These values of the Ratio, r, correspond to probabilities, P, which extend from 99% to 1%. Now you might choose as ‘uncertain’ a different range of probabilities, for example you might choose 0.1 to 10 as your range for r. These values correspond to probabilities between 90% and 10%.

Does it matter what range is taken to be ‘uncertain’?

You may remember from my ‘Philosophical bits’ that, as we collect evidence, the a posteriori probability, P, nearly always tends to either 1, total certainty, or to 0, total impossibility. There are a few scenarios in which the a posteriori probability tends to values other than 0 or 1; I may describe some of these another day. In real life such scenarios are rare and Spam detection is not one of these rare cases.

If you collect enough information (in this case by analysing ‘enough’ words from the email) then it is almost certain that the Probability returned by Bayes’ Formula will be outside this ‘uncertain’ range. So my answer to the question “Does it matter?” is: “No, it doesn’t matter what range you take to be ‘uncertain’ so long as it is a ‘reasonable’ range”! I shall leave the definition of ‘reasonable’ to you. Does it matter? Usually it doesn’t!

Emails or Words?

I have already had some correspondence about SpamStamp. A common misunderstanding is to believe that SpamStamp counts the number of emails which are Spam in order to calculate the value of r. The numbers which go into the calculation of r (and hence P) are ‘numbers of words’ and not ‘number of emails’.

Let me clarify with an example.

If the word “feet” occurs three times in one single email which is classed as Spam, then the number corresponding to the “Occurrences” of “feet” (see the screenshot of the PipeDream spreadsheet [Prob02]) will increase by three (for example, from 32 to 35) and the “Spam Dictionary Size” (see screenshot) will also increase by three (for example to 412003).

Linked Words

Next question.

The answer to this question is: “The probability associated with every individual word in the database is considered to be ‘Statistically Independent’ of the occurrence of any other word in the same email”. You have to guess what the question was! What this assumption of ‘Statistical Independence’ means is that we do not involve ourselves in the complexities which would arise if we considered these (difficult to assess) conditional probabilities.

We all know that if an email contains the word “Viagra” then the probability of that same email containing the word “Performance” will be higher than the probability of “Performance” appearing in an email which does not contain “Viagra”. This sort of conditional probability is ignored by SpamStamp. Undoubtedly there is some degree of correlation between the inclusion of ‘key words’ such as “Viagra” and ‘insignificant’ words such as “Performance” which might appear in an email about a sporting or musical event.

Epistemology

The justification for this simplification relies on the way in which Bayesian Epistemology ‘works’. Let me try to make the (philosophical) point crystal clear with a two-liner:

We are not trying to determine an accurate value for P.
We are trying to decide whether the email is Spam or Ham.

If we believe in Bayesian Epistemology (that is the use of Bayes’ Theorem as a means of increasing our ‘knowledge’) then we shall believe that by analysing enough Events (words in the email) the Probability of the email being Spam will fall well outside whatever range of values we choose to regard as ‘uncertain’!

By ignoring the conditional probabilities associated with the linking of two (or more) words such as “Viagra” and “Performance” we accept that we may arrive at the wrong value for P (the probability that the email is Spam). What we believe will happen is that the final value of P is so close to either zero or unity (impossibility or certainty) that ‘fine tuning’ this probability calculation by introducing these conditional probabilities will not influence our decision about the nature of the email.

Missing Words – 1

Suppose we have an email we are trying to assess, and it contains the word “Viagra”. Furthermore let us assume that “Viagra” appears in the the Spam database but does not appear at all in the Ham database.

In such a case, the value of r_n (check with the formula above) for this word will be $r_n = 0$. If any one of the r_n values is 0 then the value of the product of all the chain, $r = 0$. When $r = 0$ then $P = 1$, certainty. We can not really allow one word (even if it is “Viagra”) to dominate the chained calculation in this way.

In order to avoid this situation, Jan–Jaap has adjusted his program so that if the calculated value of r_n is less than 0.001 then r_n is included in the calculation as $r_n = 0.001$.

Similarly, if a word appears in the Ham database but not in the Spam database the value of r_n used in the calculation is $r_n = 1000$ rather than r = infinity (which computers don't like).

Detailed Balancing

For this simple exercise I would like you to consider an email which contains the two words "Viagra" and "Jan–Jaap". Furthermore I would like you to consider the situation where "Viagra" appears only in the Spam database and "Jan–Jaap" appears only in the Ham database. With some licence to choose my own suffices (and leave you to guess what my notation means) this would give $r_V = 0.001$ and $r_J = 1000$. What is the overall effect of the inclusion of these two words (once each) in an email?

Let's do a bit of arithmetic. We know the values of r are 'chained' together so the overall effect of "Viagra" + "Jan–Jaap" can be expressed mathematically as: $r_{VJ} = r_V * r_J$. What is this product? The answer is: $r_{VJ} = r_V * r_J = 1$.

Multiplying any number by 1 does not change it. If all the other values of r_n are known and multiplied together then multiplying this number by 1 will not change it. The effect of one 'Spam only' word together with one 'Ham only' word is 'no effect' That is because the r values of the two words balance each other out.

Those of you who feel like investigating the mathematics further might like to consider what would have happened if Jan–Jaap had not 'adjusted' extreme values of r_n to the range 0.001 to 1000. Without this truncation $r_V = 0$ and r_J = infinity.

What is the result of multiplying zero by infinity? Not something suitable for a computer! Neither is it something which is easy to calculate manually!

The Bias Factor

In Jan–Jaap's article he introduces a "bias factor" in his equation.
What this does is to change the equation $P_n = 1/(1 + r_n)$ to $P_n = 1/(1 + 2*r_n)$.

Jan–Jaap's version of SpamStamp does not use every word in the email because sufficiently significant values of P can be calculated without going to that extreme. Indeed, Jan–Jaap's program uses only the fifteen 'most significant' words in the email. (We shall see what 'most significant' means later.) For every one of the fifteen words an additional factor of 2 is introduced into the product chain used to calculate r.

Another way of looking at this is to calculate $r = r_1 * r_2 * ...$ in the usual way and then multiply the result by 2^15. 2^15 is approximately 32 000. The full Bias Factor 'shifts' the symmetrical graph of P against r along the r axis by a factor of about 32 000.

The size of this factor gives you some idea of the size of the factor needed to generate a significant amount of 'Bias'. I shall come back to considering Bias and 'factors' next month; in the meantime let's move on to another practical consideration.

Missing Words – 2

I have dealt with the case where a word is included in one of the databases but not the other. Now let's see how the program deals with words in the email which appear in neither database.

The missing word is allocated an 'inverse Bayesian probability' of $P_n = 40\%$. If we convert this to a value of r_n using $r_n = (1/P_n) - 1$ then this works out to be $r_n = 1.5$.

Words in the email which do not appear in either database have $r_n = 1.5$. If none of the words in an email appear in either database then the 'factor' will be 1.5^15. This is an r value of about 438, giving a probability, P, that the email is Spam of less than 0.25%.

By the way I have noticed that 'nonsense' words (such as "dlenciuqn", consisting of random characters) appear regularly in the subject line Spam emails and sometimes in the body of the text. If such an email is classified as Spam then these nonsense words get added to the Spam database diluting the significance of real Spam words such as "Viagra".

Summary

The version of Bayes' Equation which is most applicable to chained events is $P = 1/(1 + r)$.

A range of 'uncertain' values of P such as 99% to 1% (or 90% to 10%) can be converted into a range of values of r such as 0.001 to 1000 (or 0.01 to 100).

Nearly all 'real life' scenarios are such that, given sufficient events for analysis, the value of P tends to either zero or one (impossibility or certainty). There are some scenarios in which P tends to some intermediate value (such as 0.5).

In the case of SpamStamp, the analysis of fifteen events (the fifteen most significant words in the email) will lead to a calculated probability (that the email is Spam), P, either close to zero or close to one.

The analysis used by SpamStamp contains approximations and assumptions which detract from arriving at the most accurate value possible for P but (as in most problems which invoke Bayesian Epistemology) finding P is only a means to an end. The 'end' is to decide whether to treat the email as Spam or Ham. These approximations and assumptions do not detract from such a conclusion because the calculated value of P will be well outside the chosen 'range of uncertainty'.

Communication

Please contact me by email (preferred) or by letter if you have any questions or comments. Please email me at <Archive@abacusline.demon.co.uk> and not <gerald@abacusline...>.