Evaluating the Use of Bayes' Theorem in Spam Filtering

INTRODUCTION

We live in a world of great technological advancements ranging from communication to precise machinery. Different modes of communication including texting, calling, and email are used in a day to day basis. I am captivated by the vast use of mathematics in emails, especially spam emails. This admiration was derived from an instance that occurred in the past with my father. When email was first introduced in the 1970's[1], the filtering methods for spam were not as effective as they are now. My father had bad luck with spam emails in the past as one of his emails mistakenly went into the spam folder. This caused my dad to ignore an important work email which then caused numerous issues. This made me explore different spam filters and evaluate its effectiveness.

PURPOSE

This investigation will evaluate a widely used statistical spam filter, the Bayesian spam filter. This filter uses Bayes' theorem in discrete mathematics to determine if an incoming email is considered to be spam or ham (term used for an email that is not spam). Bayes' theorem (aka Bayes' rule) is a formula that determines the probability of an event (A), given that another event (B) has already occurred[2]. Mathematically[2]:

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A) * P(B \mid A)}{P(B)} \quad P(B) \neq 0$$

Where:

- $P(A \mid B)$ is the probability of observing event A, given that event B has occurred
- $P(A)$ *and* $P(B)$ are the probabilities of A and B without observing one another respectively
- $P(B \mid A)$ is the probability of observing event B, given that event A has occurred

DERIVATION OF BAYES' RULE[2]

Rule of conditional probability[3]:

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}$$

Equivalent:

$$P(B \mid A) = \frac{P(B \cap A)}{P(A)} = \frac{P(A \cap B)}{P(A)}$$

Rearranged for $P(A \cap B)$:

$$P(A \cap B) = P(B \mid A) * P(A)$$

Plug in previous expression to original expression ($P(A \mid B) = \frac{P(A \cap B)}{P(B)}$):

$$P(A \mid B) = \frac{P(B \mid A) * P(A)}{P(B)}, \; P(B) \neq 0$$

APPLYING BAYES' THEOREM TO SPAM FILTERING

There are several types of spam filters that classify emails into spam or ham. A bayesian spam filter uses the occurrence of spammy words (words that are typically found in spam emails) and hammy words (words that are typically found in ham emails) to compute the probability that an incoming email is spam. The essential detail of Bayes' rule is that each token (term used in programming meaning word) is independent. This application involves a series of three steps which will determine if an incoming email is spam or ham. The following formula applies Bayes' theorem to compute the probability of an email being spam given a specific word (w_x)[4]:

$$P(S \mid W_x) = \frac{P(W_x \mid S) * P(S)}{P(W_x \mid S) * P(S) + P(W_x \mid H) * P(H)}$$

Where:

- $P(S \mid W_x)$ is the probability that the email is spam with knowledge that it contains the given word

- $P(W_x \mid S)$ is the probability that the given word occurs in spam email

- $P(S)$ is the total probability that the email is spam

- $P(W_x \mid H)$ is the probability that the given word occurs in ham email

- $P(H)$ is the total probability that the email is ham

INTEGRATING INDIVIDUAL PROBABILITIES (P(S | W))

Once all the spam probabilities of each word in a specific email have been found, the next step is to determine the probability of the email being spam or not using the following formula[5]:

$$p_S = \frac{\prod_{x=1}^{X} p_x}{\prod_{x=1}^{X} p_x + \prod_{x=1}^{X} (1 - p_x)}$$

Where:

- Π is the product operator which is a process of multiplying numbers together. Analogous to Σ which is an operation to add numbers together.

- $p_x = P(S \mid W_x)$

- p_s is the probability that the incoming email is spam

This formula is what is known as the naive Bayes' classifier[5]. This is a number that is compared to a threshold for classification. If p_s is lower than the threshold, the email is classified as ham. If p_s is higher than the threshold, the email is classified as spam.

THRESHOLD FOR CLASSIFICATION (λ)

A threshold for classification is the final step of Bayesian spam filtering. This threshold will be denoted as lambda (λ) and is expressed as the following:

$$\frac{P_S}{P_H} > \lambda$$

Where:

- p_s is the probability that the incoming email is spam

- p_H is the probability that the incoming email is ham

- $p_s + p_H = 1$, $\therefore\ p_H = 1 - p_s$

This inequation states that the ratio of the probability that the email is spam to the probability that the email is ham must be greater than λ in order for the email to be placed in the spam folder.

Mistaking a ham email as spam is a much more severe problem than letting spam go through the filter by being mistakenly classified as ham. As stated in the introduction, this happened to my father and he had to face major consequences.

λ can be any high number that can classify whether an email is ham or spam. The higher the number, the safer. I will choose λ to be **100** which means that classifying a ham email as spam is as bad as having 100 spam emails in one's inbox. Therefore, mathematically:

$$\frac{P_S}{P_H} > 100$$

EXAMPLE ONE (SPAM EMAIL)

The following is a table that shows the training dataset acquired from my father's gmail. This information is recorded at the time of this writing:

Table 1: Total number of emails from my father's gmail account

	Ham	**Spam**	**Total**
# of emails	2271	502	2271 + 502 = 2773

The following is a screenshot of a spam email. The spam filtering system is from google and is nearly 100% correct in terms of classifying emails. Using the dataset from the appendix (p.14-15) and table 1, I will evaluate and figure out if I can verify that this email is actually considered spam using the Bayesian filtering method described above.

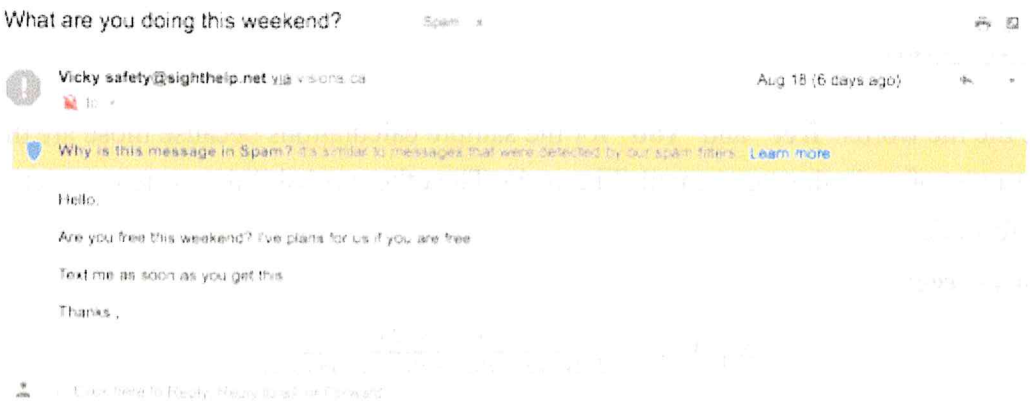

Figure 1: Spam email example.

Table 2: Each word and its respected spam probability from figure 1. (from appendix p.14-15)

w_x	**word**	$P(S \mid W_x)$
w_3	are	0.867
w_4	doing	0.925
w_7	**free**	**0.649**
w_8	for	0.875
w_{12}	get	0.788
w_{15}	hello	0.609
w_{16}	if	0.650
w_{17}	me	0.962
w_{20}	plans	0.479
w_{21}	soon	0.544
w_{23}	this	0.862
w_{26}	us	0.739
w_{31}	what	0.712
w_{32}	**you**	**0.332**

SAMPLE CALCULATIONS

I will use the the words "**free**" and "**you**" for the sample calculations because those are the words that occur most often in the spam email (Figure 1). The other probabilities were calculated using the same formula.

Step 1) w_7 **- free:**

$$P(S \mid W_7) = \frac{P(W_7 \mid S) * P(S)}{P(W_7 \mid S) * P(S) + P(W_7 \mid H) * P(H)}$$

$$P(S \mid free) = \frac{P(free \mid S) * P(S)}{P(free \mid S) * P(S) + P(free \mid H) * P(H)}$$

$$P(S \mid free) = \frac{(253/502)(502/2773)}{(253/502)(502/2773) + (137/2271)(2271/2773)}$$

$$P(S \mid free) = \frac{0.0912}{0.0912 + 0.0494}$$

$$P(S \mid free) = 0.649$$

w_{32} **- you:**

$$P(S \mid W_{32}) = \frac{P(W_{32} \mid S) * P(S)}{P(W_{32} \mid S) * P(S) + P(W_{32} \mid H) * P(H)}$$

$$P(S \mid you) = \frac{P(you \mid S) * P(S)}{P(you \mid S) * P(S) + P(you \mid H) * P(H)}$$

$$P(S \mid you) = \frac{(391/502)(502/2773)}{(391/502)(502/2773) + (786/2271)(2271/2773)}$$

$$P(S \mid you) = \frac{0.141}{0.141 + 0.283}$$

$$P(S \mid you) = 0.332$$

Step 2)

$$p_s = \frac{\prod_{x=1}^{X} p_x}{\prod_{x=1}^{X} p_x + \prod_{x=1}^{X} (1 - p_x)}$$

$$p_s = \frac{\prod_{x=1}^{14} p_x}{\prod_{x=1}^{14} p_x + \prod_{x=1}^{14} (1 - p_x)}$$

$$p_s = \frac{0.005}{0.005 + (7.94 * 10)}$$

$$p_s = 0.998$$

Step 3)

$$\frac{P_S}{P_H} > 100$$

$$\frac{0.998}{1-0.998} > 100$$

$$499 > 100$$

499 is indeed greater than 100, which therefore verifies that this email is spam!

EXAMPLE TWO (HAM EMAIL)

The following is a screenshot of a ham email. This email also has the same filter from google and is nearly 100% accurate in terms of classifying emails. Through Bayesian filtering, I will evaluate and verify that the email is actually legitimate.

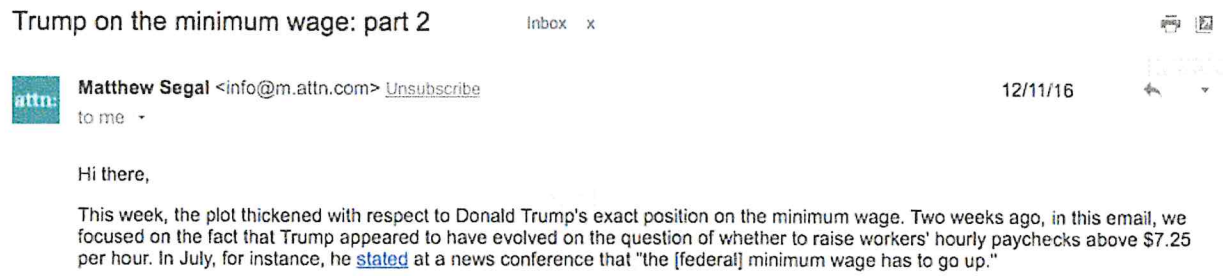

Figure 2: Ham email example.

Table 3: Each word and its respected spam probability from figure 2. (from appendix p.14-15)

w_x	**word**	$P(S \mid W_x)$
w_1	ago	0.0810
w_2	above	0.176
w_5	exact	0.123
w_6	email	0.742
w_8	for	0.875
w_9	fact	0.516

w_{10}	**federal**	**0.711**
w_{11}	go	0.766
w_{13}	hi	0.763
w_{14}	hour	0.757
w_{18}	news	0.0823
w_{19}	per	0.876
w_{22}	there	0.598
w_{23}	this	0.862
w_{24}	that	0.706
w_{25}	up	0.612
w_{27}	week	0.227
w_{28}	with	0.781
w_{29}	weeks	0.648
w_{30}	we	0.874

SAMPLE CALCULATION

I will use the word "**federal**" for the sample calculation. This is a word that is usually not associated with a spam email. The other probabilities were calculated using the same formula.

Step 1) w_{10} - federal:

$$P(S \mid W_{10}) = \frac{P(W_{10} \mid S) * P(S)}{P(W_{10} \mid S) * P(S) + P(W_{10} \mid H) * P(H)}$$

$$P(S \mid federal) = \frac{P(federal \mid S) * P(S)}{P(federal \mid S) * P(S) + P(federal \mid H) * P(H)}$$

$$P(S \mid federal) = \frac{(34/502)(502/2773)}{(34/502)(502/2773) + (13/2271)(2271/2773)}$$

$$P(S \mid federal) = \frac{0.0123}{0.0123 + 0.005}$$

$$P(S \mid federal) = 0.711$$

Step 2)

$$p_S = \frac{\prod_{x=1}^{X} p_x}{\prod_{x=1}^{X} p_x + \prod_{x=1}^{X} (1 - p_x)}$$

$$p_S = \frac{\prod_{x=1}^{18} p_x}{\prod_{x=1}^{18} p_x + \prod_{x=1}^{18} (1 - p_x)}$$

$$p_S = \frac{5.78 * 10^{-7}}{5.78 * 10^{-7} + 3.93 * 10^{-9}}$$

$$p_S = 0.993$$

Step 3)

$$\frac{P_S}{P_H} > 100$$

$$\frac{0.993}{1-0.993} > 100$$

$$142 > 100$$

The email is said to be in the inbox. However, 142 is greater than 100 which indicates that the email is spam! This means that the Bayesian spam filter produced a false positive which is this case caused this legitimate email to be placed in the spam folder.

CONCLUSION

From carrying out this evaluation, I can draw some significant conclusions regarding a Bayesian spam filter. Firstly, the Bayesian spam filter does not have a success rate of 100%. This will be discussed in more detail. In general, Bayes' theorem treats each word as independent and only focuses on the words, not phrases or pictures which thus causes problems.

Appointing words as tokens and using Bayes' theorem gave the following formula:

$$P(S \mid W_x) = \frac{P(W_x \mid S) * P(S)}{P(W_x \mid S) * P(S) + P(W_x \mid H) * P(H)}$$

By determining the spam probability of each word, the following formula was derived using Bayes' rule:

$$p_S = \frac{\prod_{x=1}^{X} p_x}{\prod_{x=1}^{X} p_x + \prod_{x=1}^{X} (1 - p_x)}$$

Using the threshold for classification being $\lambda = 100$, any incoming email was determined to be spam through the following constraint:

$$\frac{P_S}{P_H} > 100$$

From the first example, the spam email was indeed verified through the Bayesian spam filter. This was done using the primary dataset from the tables and the appendix which then gave a correct classification. My father now uses a spam filter that rarely produces a false positive or false negative. This means that the statistical method works for emails that have few repeated words most often used in spam emails (figure 1). However, the second email gave the wrong conclusion as it was classified as spam through the Bayesian filter. The context of the ham email (figure 2) indicates that it shouldn't be a spam email. However, Bayes' rule keeps the words independent with no context. The words in the email have a high spam probability like "for" and "per". This is what is known as Bayesian poisoning which in turn reduces the effectiveness of Bayesian spam filtering.

Other spam filters include header based filters and a permission filter[6]. These filters are successful, however, they take time to adapt to a large multitude of different words in every email. As I was looking through all these filters, I found that the statistical approach would be

the most successful as it is adaptive to a particular organization's needs. In order for spammers to trick the Bayesian filter, they need to know inside information about the organization which is highly unlikely and difficult. However, clever spammers can use words that have a low spam probability which then causes Bayesian poisoning discussed above. In addition, a Bayesian spam filter has a slow learning rate (2 weeks)[7] which also reduces the effectiveness of Bayesian spam filtering.

This investigation can be improved by adding more examples of emails that contain rare words. Rare words are commonly used by spammers as they confuse the filter. Thus, the Naive Bayes' classifier would include more variables as more words outside my training dataset are used in the emails. Gmail uses a complex spam filter that is closely related to the Bayesian spam filter[8]. Google also uses a three step process in the classification of emails. First, each word in an email has an associated numerical value which is based on the likelihood that the email is spam. Then, a complex equation is used that takes into account rare words, pictures, and videos. Finally, the output value of the equation is tested through a sensitivity threshold, similar to my threshold of classification, and thus is categorized as spam or ham. According to google, their spam filter has a 99.9% success rate.[9]

For further direction, the Bayesian spam filter should be capable of processing images in the message as many spam emails attach photos. In addition, the spam probability should not only be calculated through words, but also phrases. Finally, the spam filter should operate with any email of a different language. This can ensure more safety for anyone that uses the email, including my father.

REFERENCES

- [1]History of Email. (n.d.). Retrieved August 21, 2017, from http://www.inventorofemail.com/history_of_email.asp

- [2]Staff, I. (2009, October 02). Bayes' Theorem. Retrieved July 19, 2017, from http://www.investopedia.com/terms/b/bayes-theorem.asp

- [3]Conditional Probability. Retrieved August 02, 2017, from http://www.stat.yale.edu/Courses/1997-98/101/condprob.htm

- [4]Anderson, S. D. (n.d.). Combining Evidence using Bayes' Rule (pp. 01-04, Publication). Retrieved from http://cs.wellesley.edu/~anderson/writing/naive-bayes.pdf

- [5]Graham, P. (n.d.). Better Bayesian Filtering (Tech.). Retrieved August 07, 2017, from http://www.paulgraham.com/naivebayes.html

- [6]B. (2009, February 04). Most Common Spam Filter Triggers [Web log post]. Retrieved August 09, 2017, from https://blog.mailchimp.com/most-common-spam-filter-triggers/

- [7]Provost, J. (n.d.). Na¨ıve-Bayes vs. Rule-Learning in Classification of Email (pp. 01-04, Rep.). Retrieved August 23, 2017, from ftp://ftp.cs.utexas.edu/pub/AI-Lab/tech-reports/UT-AI-TR-99-284.pdf.

- [8]Eveleth, R. (2012, October 03). How Google Keeps Your Spam Out of Your Inbox. Retrieved September 11, 2017, from https://www.smithsonianmag.com/smart-news/how-google-keeps-your-spam-out-of-your-inbox-58828900/

- [9]Metz, C. (2017, June 03). Google Says Its AI Catches 99.9 Percent of Gmail Spam. Retrieved September 16, 2017, from https://www.wired.com/2015/07/google-says-ai-catches-99-9-percent-gmail-spam/

- Eberhardt, J. J. (n.d.). Bayesian Spam Detection (pp. 01-05, Tech.). Retrieved July 29, 2017, from http://digitalcommons.morris.umn.edu/cgi/viewcontent.cgi?article=1024&context=horizons

- Zdziarski, J. (2005). Ending spam Bayesian content filtering and the art of statistical language classification. San Francisco: No Starch Press.

- Good, I. (1965). The estimation of probabilities; an essay on modern Bayesian methods. Cambridge: Mass.

- Bessière, P. (2014). Bayesian programming. Boca Raton: Taylor & Francis.

APPENDIX

The following is a set of words used in the Bayesian filter. My **primary data set** has a total of 2271 ham emails and 502 spam emails (table 1), each word shown in every email has been calculated. This was done by copying all ham and spam emails and pasting it onto a document. The specific words selected for this internal assessment were the words used in the two examples shown above. The total amount of each specific word was simply calculated by the word count of a specific word on the document. Each row displays the word, the number of spam emails that contain that word, and the number of ham emails that contain that word.

w_x	Word (w)	# of spam emails with w_x	# of ham emails with w_x
w_1	ago	37	161
w_2	above	83	252
w_3	are	286	1422
w_4	doing	57	75
w_5	exact	58	45
w_6	email	143	487
w_7	free	253	137
w_8	for	378	1829
w_9	fact	50	167
w_{10}	federal	34	13
w_{11}	go	115	130
w_{12}	get	211	235
w_{13}	hi	33	39
w_{14}	hour	68	55

w_{15}	hello	47	16
w_{16}	if	281	826
w_{17}	me	113	447
w_{18}	news	35	37
w_{19}	per	99	155
w_{20}	plans	34	22
w_{21}	soon	62	124
w_{22}	there	172	783
w_{23}	this	329	1317
w_{24}	that	262	1215
w_{25}	up	178	372
w_{26}	us	190	467
w_{27}	week	98	86
w_{28}	with	294	1328
w_{29}	weeks	64	99
w_{30}	we	248	712
w_{31}	what	178	545
w_{32}	you	391	786