HUMBOLDT-UNIVERSITÄT ZU BERLIN

A Bayesian E-Mail Spam Filter

Sabine Bertram, Carolina Gumuljo,
Sophie Stadlinger, Karolina Stańczak

Department of Computer Science,
Machine Learning Group

Contact Information:
sabine.bertram@mailbox.org
carolina.gumuljo@student.hu-berlin.de
s.stadlinger1@gmx.de
kstanczak@gmail.com

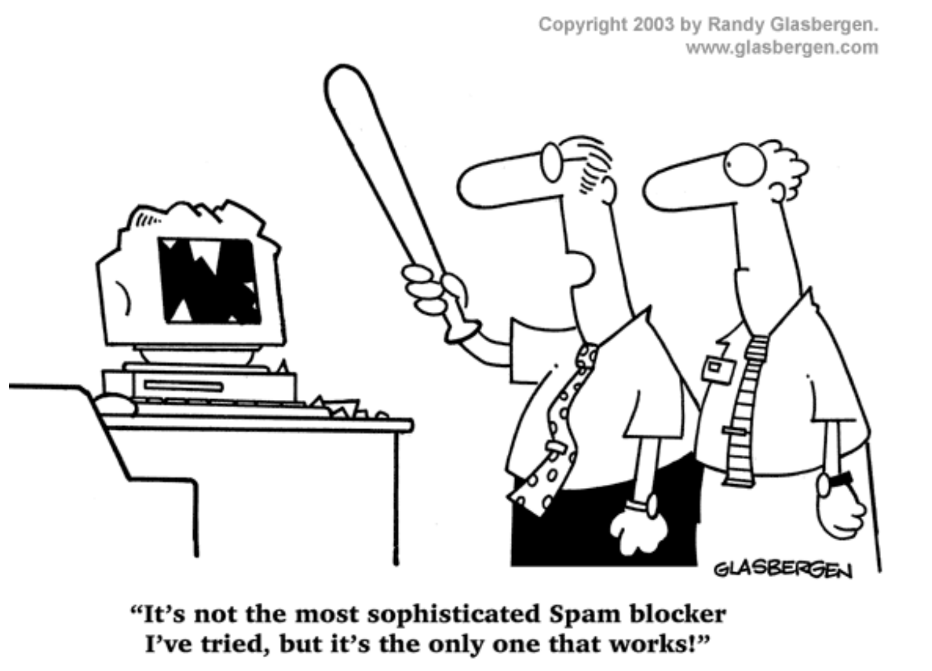

Copyright 2003 by Randy Glasbergen.
www.glasbergen.com
GLASBERGEN
"It's not the most sophisticated Spam blocker I've tried, but it's the only one that works!"

Figure 1: Cartoon © Randy Glasbergen, used with special permission from www.glasbergen.com.

Abstract

The goal of our project was to construct a Bayesian E-Mail Spam Filter which accurately classifies emails in Spam and Ham (=no spam). Comparing different Bayesian approaches with varying methods to combine the feature probabilities, the Fisher method yields the most accurate predictions.

Introduction

With emails having become one of the main communication tools, good email spam filters have increasingly gained importance. One very efficient way to classify emails is the Bayesian approach, which Graham [3] and Robinson [5] described and developed in the early 2000s. Based on their approach we build such a filter to accurately distinguish between Spam and Ham emails.

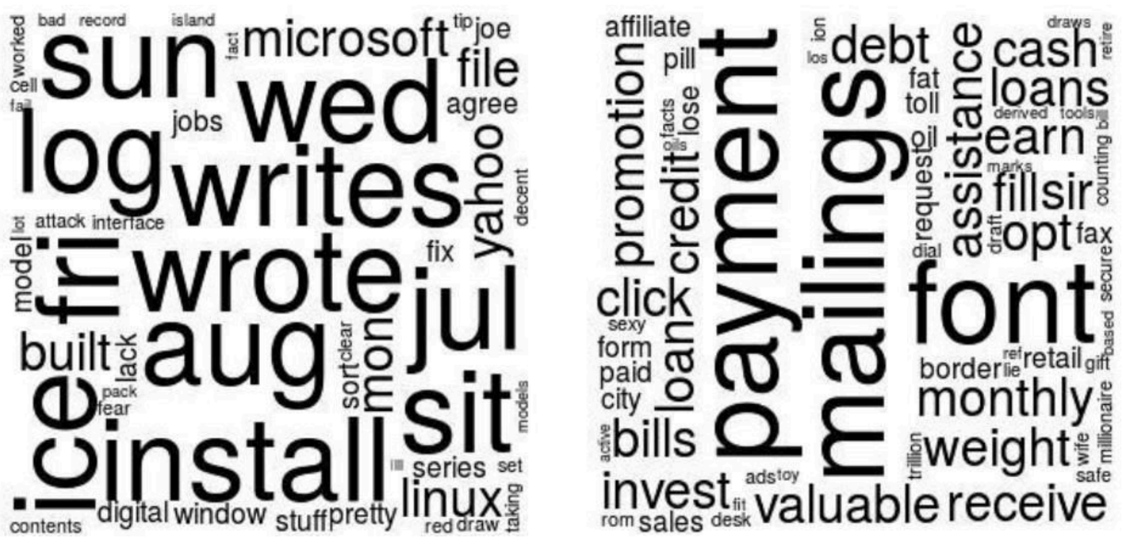


Figure 2: Clouds for ham (left) and spam (right) words.

Data Preprocessing

The Bayesian Spam filter was built by using the SpamAssassin training data. After constructing a large character object for spam and ham emails, respectively, and splitting these data sets into training and test sets, the raw training data was pre-processed by using the R package `tm`. Next, term document matrices were created containing term frequencies per email for each training set. Additionally, we matched the terms with an English and a German dictionary.

Model Building

As our baseline for comparison we constructed a Naive Bayes Classifier. When employing Naive Bayes, we use the relative frequency of spam as the prior. Another often used prior is the so called *Beta prior* (see [4]) which belongs to the family of conjugate priors. The Beta prior is defined by[1]

$$p(\theta) = B(\theta|\alpha_s, \alpha_h) = \frac{\Gamma(\alpha_s + \alpha_h)}{\Gamma(\alpha_s)\Gamma(\alpha_h)}\theta^{\alpha_s - 1}(1-\theta)^{\alpha_h - 1} \tag{1}$$

The hyperparameters can be interpreted as being previously observed pseudo-data, namely α_s spam and α_h ham emails out of a pseudo-sample of size $\alpha_s + \alpha_n$. It can be shown that the posterior distribution is again a Beta distribution, defined as follows[2]:

$$p(\theta|w) = Beta(\theta|\alpha_s + n_s, \alpha_h + n_h) = \frac{\Gamma(\alpha_s + \alpha_h + N)}{\Gamma(\alpha_s + n_s)\Gamma(\alpha_h + n_h)}\theta^{\alpha_s + n_s - 1}(1-\theta)^{\alpha_h + n_h - 1} \tag{2}$$

The probability of the next email being spam is the expected value over the posterior distribution:

$$p(X_{N+1} = Spam|w) = \int p(X_{N+1} = Spam|w)p(\theta|w)d\theta = \int \theta p(\theta|w)d\theta = \mathbb{E}_{p(\theta|w)}(\theta) \tag{3}$$

For the Beta distribution, this yields to the expression

$$p(X_{N+1} = Spam|w) = \mathbb{E}_{p(\theta|w)}(\theta) = \frac{\alpha_s + n_s}{\alpha_s + \alpha_h + N} \tag{4}$$

To build our spam filter, we first calculated the relative frequencies of a word w in the corpus of spam (b) and ham (g) emails. Then, we computed $p(w) = \frac{b(w)}{b(w)+g(w)}$, which can be interpreted as the probability that a randomly chosen email containing word w will be spam. If we define $s = \alpha_s + \alpha_h$ and $x = \frac{\alpha_s}{s}$, the probability that the next email containing word w is spam, is defined as[3]

$$p(X_{N+1} = Spam|w) = \frac{s \cdot x + n_s}{s + N} \tag{5}$$

Approximating n_s by $N \cdot p(w)$ yields Robinson's (2003) formula

$$p(X_{N+1} = Spam|w) = f(w) = \frac{s \cdot x + N \cdot p(w)}{s + N} \tag{6}$$

Combining the Feature Probabilities and Classification

To get from $p(X_{N+1} = Spam|w)$ to $p(X_{N+1} = Spam|\mathbf{w})$, one can take the product of the single probabilities $p(X_{N+1} = Spam|w_i)$, i.e.,

$$p(X_{N+1} = Spam|\mathbf{w}) = \prod_{i=1}^{M} p(X_{N+1} = Spam|w_i) = \prod_{i=1}^{M} f(w_i) \tag{7}$$

where M is the number of words in email X_{N+1}. But the assumed independence of these probabilities is quite arguable. Thus, Robinson [5] defines $H = C^{-1}(-2ln\prod_{i=1}^{M} f(w_i), 2M)$ and $S = C^{-1}(-2ln\prod_{i=1}^{M}(1 - f(w_i)), 2M)^4$ for combining probabilities given that any dependence, if existing, is taken into account. In order to finally classify the mails into Spam or Ham, we use the index $I = \frac{1+H-S}{2}$ and say that an email is spam if $I > 0.5$ and ham if $I < 0.5$ (cf. [5]).

Results

Method	s_{opt}	x_{opt}	$\mathrm{F1}_{Spam}$	p_{Spam}	r_{Spam}	$\mathrm{F1}_{Ham}$	p_{Ham}	r_{Ham}
Fisher	10	0.18	95.70%	93.68%	97.80%	98.09%	99.04%	97.17%
Stouffer	11	0.38	63.38%	72.28%	56.44%	79.59%	74.48%	85.45%
Mudholkar	10	0.18	95.39%	93.68%	97.09%	97.93%	98.72%	97.16%

Table 1: Hyperparameters, F1 score and its components for different combination methods

- The Fisher method is the best way to combine single word probabilities, regardless which optimization strategy is followed.
- The Mudholkar method produces almost as good results for both optimization strategies.
- The Stouffer method fails to even outperform the Naive Bayes classifier which yields 87.08% for ham emails and 75.32% for spam emails, consdering the F1 Score.

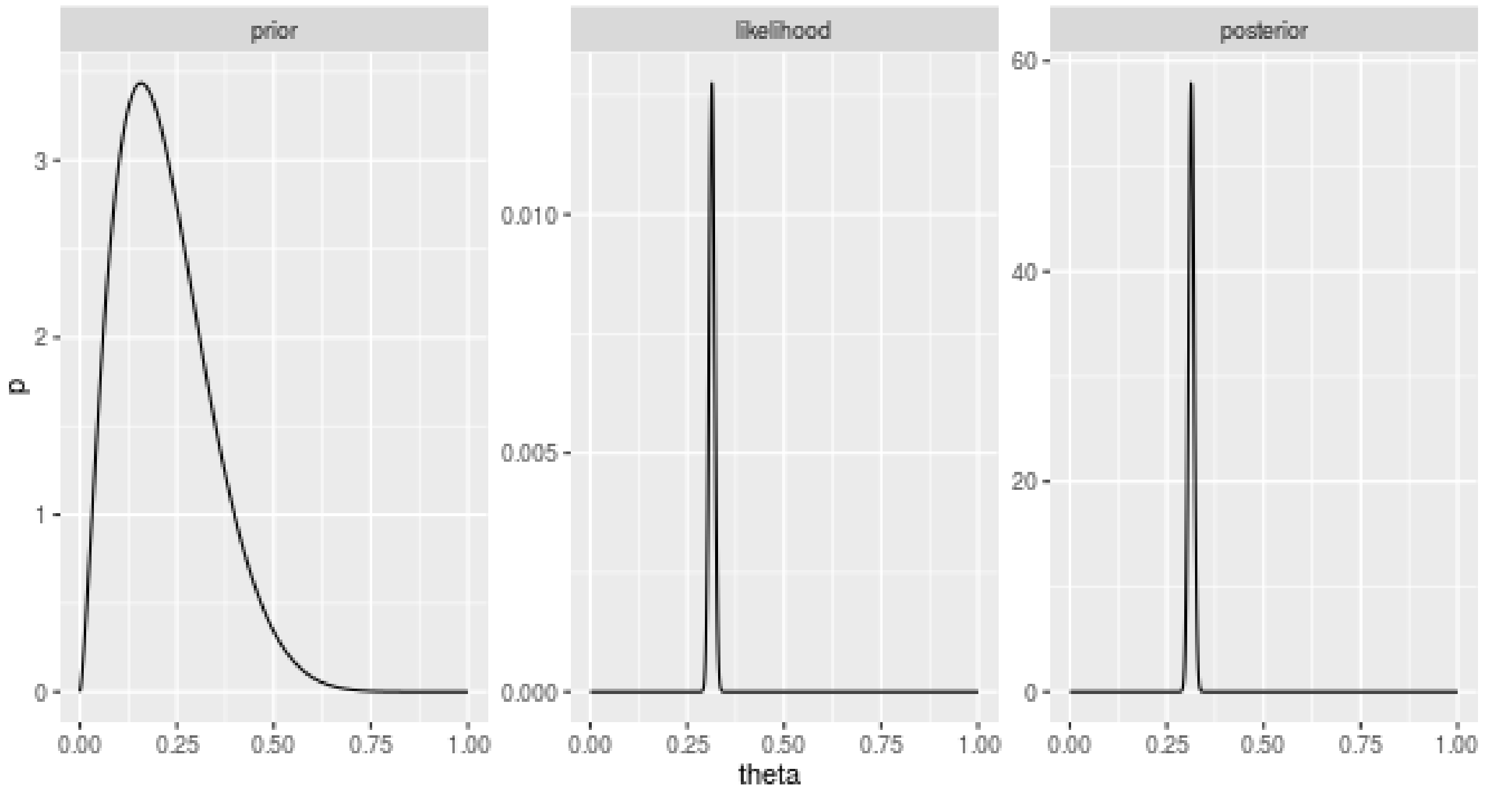


Figure 3: Prior, likelihood and posterior distribution for F1 score

Conclusion

Comparing the different spam filters, the Fisher combination method seems to be a slightly better approach for our data split, yielding 93.68% correctly classified Spam emails and 99.04% correctly classified Ham emails (for F1 optimization). In order to get more reliable results, a cross-validation should be conducted. For further research, an expanding training set should be used to improve the classifier.

References

[1] Conway, Drew and John M. White (2012): *Machine Learning for Hackers*. O'Reily and Associates, 1st edition.

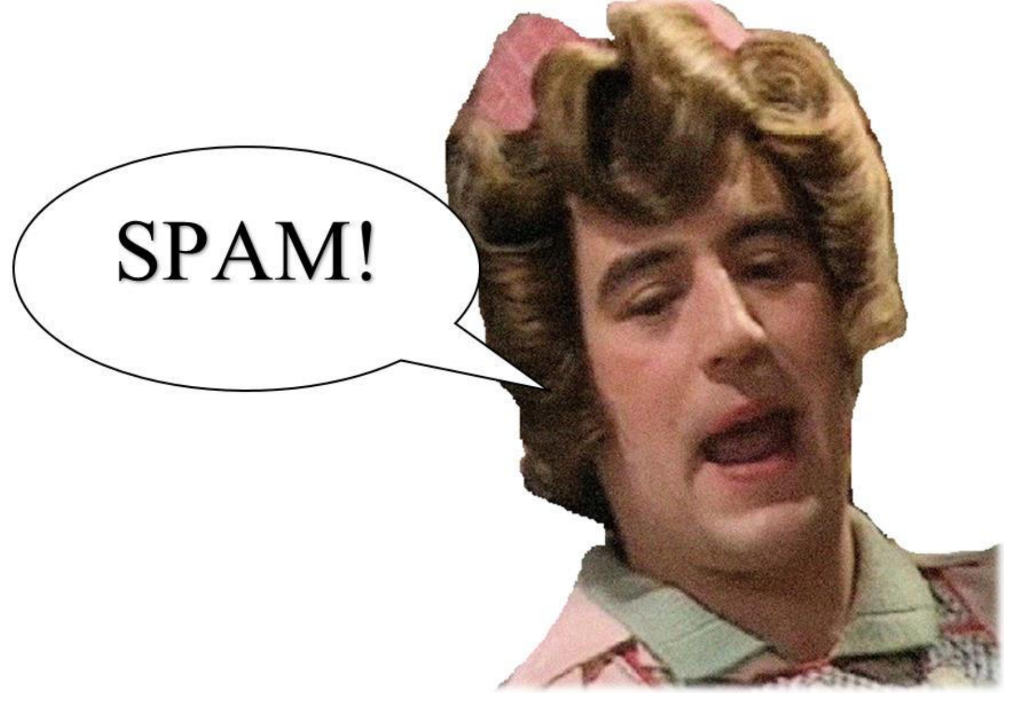
[2] Folks, J. and R. Little (1971): *Asymptotic Optimality of Fisher's Method of Combining Independent Tests*. Journal of the American Statistical Association, 66, pp. 802-806.

[3] Graham, Paul (2002): *A Plan for Spam*, available online on: `http://www.paulgraham.com/spam.html` (last retrieved on July 14, 2016).

[4] Heckerman, David; Holmes, D. E. and L. C. Jain (Eds.): *A Tutorial on Learning with Bayesian Networks*, In: Innovations in Bayesian Networks: Theory and Applications, Springer Berlin Heidelberg, 2008, pp. 33-82.

[5] Robinson, Gary (2003): *A Statistical Approach to the Spam Problem*, available online on: `http://www.linuxjournal.com/article/6467` (last retrieved on July 14, 2016).

[6] Winkler, Anderson (2016): *Non-Parametric Combination (NPC) for brain imaging*, available online on: `https://brainder.org/2016/02/08/npc/` (last retrieved on July 14, 2016).



[1] $\Gamma(\cdot)$ is the Gamma distribution and $\alpha_s > 0$ and $\alpha_h > 0$ are hyperparameters.
[2] w is a single word, n_s and n_h are the number of spam and ham emails, respectively, in the sample of sample size $N = n_s + n_h$.
[3] s is the size of our pseudo-sample and can be interpreted as the strength we give to our background information and x is the fraction of spam emails in our pseudo-sample and can be interpreted as the probability that a word we do not have any other experience of will first appear in a spam email. n_s is the number of spam emails containing word w.
[4] C^{-1} is the inverse χ^2 distribution with $2M$ degrees of freedom. This procedure was established by R. A. Fisher and its optimality was proven by Folks and Little (1971).