Humboldt-Universität zu Berlin
Lehrstuhl für Maschinelles Lernen
Sommersemester 2016
Maschinelles Lernen 1

Project Report:

"Bayesian E-Mail Spam Filter"

The Bayesians
Sabine Bertram, Carolina Gumuljo, Sophie Stadlinger, Karolina Stanczak
sabine.bertram@mailbox.org, carolina.gumuljo@student.hu-berlin.de,
s.stadlinger1@gmx.de, kstanczak@gmail.com
Master Statistik, 4. Fachsemester
Matrikelnummern: 564469, 566179, 564416, 526005
July 8, 2016

Contents

1 Introduction

Following a Bayesian approach which Graham (2002) and Robinson (2003) described and developed in the early 2000s, the goal of our project was to build a Bayesian classifier that accurately categorizes emails in "Spam" or "Ham" (= no spam). To fulfill this task, we received the SpamAssassin training data which contained emails sorted in folders named "Spam" and "Ham".

2 Data Preprocessing

For the construction of a suitable framework for our analysis, we constructed two large character objects ham and spam with the individual emails as different observations as described in Conway and White (2012). For later evaluations, each of the data sets was split into a training, a validation and a test set[1]. Using the R package `tm`, the raw train data sets were pre-processed (e.g. setted to lower case, removed from whitespaces, etc.). Next, term document matrices were created containing term frequencies per email for each training set. The underlying idea is that emails containing a lot of "spam words" will probably be spam and emails with many "ham words" are likely to be ham. To avoid sparsity of the matrices and to clean them of useless terms, e.g. resulting from typing errors, we matched the terms with an English and a German dictionary[2].

3 Model Building

3.1 Classification Based on a Beta Prior

As our baseline for comparison we constructed a Naive Bayes Classifier. When employing Naive Bayes[3], we use the relative frequency of spam as the prior. Another often used prior is the so called *Beta prior* (see Heckerman (2008)) which belongs to the family of conjugate priors, meaning the posterior will follow the same distribution

[1] These three sets were fixed by a seed of 16 due to our limitation of computation resources. All further references to accuracy are based on this particular split. In order to make general statements on the performance of a procedure, a cross validation should be conducted.

[2] To get an overview of our train data sets, we constructed dataframes containing the terms, the accumulated term frequencies, the document frequencies corresponding to each term (occurrences), the relative frequency of each term (density) and the term-frequency-inverse-document-frequency. For classification, however, only the term frequencies were needed and they were computed for each email in the test sets individually.

[3] Detailed description in 7.1.

as the prior. The Beta prior is defined by[4]

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

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 $\alpha_s + \alpha_n$.

As mentioned above, it can be shown[5] that the posterior distribution is again a Beta distribution, defined as follows[6]:

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

Coming back to our goal of classification, 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[7]

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

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

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

[4] $\Gamma(\cdot)$ is the Gamma distribution and $\alpha_s > 0$ and $\alpha_h > 0$ are hyperparameters

[5] Proof in Appendix 7.2

[6] 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 $N = n_s + n_h$.

[7] 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.

3.1.1 Finding the Hyperparameters

To find the optimal hyperparamaters we set up a grid search starting with $s = [1, 3, 5, 7, 9, 11]$ and $x = [0.2, 0.3, 0.4, 0.5]$. After the predictions for all 24 combinations were computed, the accuracy on the validation set was measured[8]. Then the hyperparameters for the two best classifications were compared. If they were the same, the grid was made finer by constructing an interval around them. If they were unequal, values in between them were taken to make the grid more precise. Then, a second grid search was conducted.

3.1.2 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 simply 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}. However, this approach assumes independence of these probabilities which is quite arguable. Thus, Robinson (2003) suggests the following procedure. He defines $H = C^{-1}(-2ln\prod_{i=1}^{M} f(w_i), 2M)$ and $S = C^{-1}(-2ln\prod_{i=1}^{M}(1 - f(w_i)), 2M)$[9]. It is the most common way of combining probabilities and ensures that any dependence, if existing, is taken into account[10].

The final step is to classify the mails into spam or ham. We could compare the results for H and S and classify an email as being spam if $H > S$ and as being ham if $H < S$. Robinson (2003), however, introduces an index $I = \frac{1+H-S}{2}$ and says that an email is spam if $I > 0.5$ and ham if $I < 0.5$. This procedure is a little more intuitive.

4 Model Evaluation

In order to measure the accuracy of our predictions, we implemented two different methods. PCC simply measures the percentage of correctly classified cases, which is defined as $\text{PCC} = \frac{tp}{tp+fp}$, where tp and fp stand for *true positives* and *false positives*, respectively. Intuitively, we would like to maximize the the PCC for ham emails, because it is more important to receive all important mails than to have an inbox without any spam. The second method is the F1 score, which takes exactly this intuition into account. The accuracy measure $\text{F1} = 2 \cdot \frac{p \cdot r}{p+r}$ weights precision p and

[8] For more details refer to Section 4.

[9] 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).

[10] For other ways to combine the probabilities see Section 7.3.

recall r equally, where p is equivalent to the PCC ($p = \frac{tp}{tp+fp}$) and $r = \frac{tp}{tp+fn}$. Here, fn stands for *false negatives*[11]. Both measures were applied to the test set.

5 Results

Comparing results, the Naive Bayes approach resulted in 82.46% correctly classified spam emails and 83.31% correctly specified ham emails. The F1 score yields 87.08% for ham emails but only 75.32% for spam emails. The Fisher method is the best way to combine single word probabilities, regardless which optimization strategy is followed (see Tables 1 and 2). When maximizing towards PCC, it yields the highest percentages correctly classified for both spam (93.68%) and ham (99.04%), and when considering the F1 score, it yields the highest values (95.70% and 98.09% for spam and ham, respectively) compared to the other methods[12]. However, the Mudholkar method produces almost as good results for both optimization strategies. Interestingly, the Stouffer method fails to even outperform the Naive Bayes classifier. Also surprising is the fact that the classification does not change with the optimization strategy on this certain data split, even though the hyperparameters change at least for the Fisher method.

5.1 Credible Intervals

For the Fisher combination method as our most effective classifier, we constructed credible intervals[13]. Regarding the optimization towards the PCC, this yields a 95% credible interval of [0.2996163; 0.3270932]. Considering a maximization of the F1 Score for spam emails, the 95% credible interval is given by [0.2995952; 0.3270747]. The corresponding graphs can be found in the Appendix.

6 Conclusion

Comparing the different spam filters, the Fisher combination method seems to be a slightly better approach for our data split. However, in order to make valid statements about the best combination method and the set of hyperparameters, a cross-validation should be conducted. Additionally, it would be interesting to see how the classifier improves if one adds test emails to the training data set when they were correctly classified. However, this is left for further research.

[11] More intuition behind these two measures is given in 7.4

[12] Note that PCC in Table 1 corresponds to p in Table 2

[13] The bayesian credible interval for a beta-distributed posterior is defined as $[q_{Beta}(0.025|\alpha_s + n_s, \alpha_h + n_h), q_{Beta}(0.975|\alpha_s + n_s, \alpha_h + n_h)]$, where q_{Beta} is the quantile function of the Beta distribution.

References

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

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.

Graham, Paul (2002): *A Plan for Spam*, available online on: http://www.paulgraham.com/spam.html (last retrieved on July 8, 2016).

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.

Robinson, Gary (2003): *A Statistical Approach to the Spam Problem*, available online on: http://www.linuxjournal.com/article/6467 (last retrieved on July 8, 2016).

Winkler, Anderson (2016): *Non-Parametric Combination (NPC) for brain imaging*, available online on: https://brainder.org/2016/02/08/npc/ (last retrieved on July 8, 2016).

7 Appendix

7.1 Naive Bayes Classifier

The baseline for comparison was a Naive Bayes classifier, a simple probabilistic classifier relying on strong independence assumptions. Based on Bayes' formula, the probability for an email to be spam can be written as

$$p(\theta|\mathbf{w}) = \frac{p(\mathbf{w}|\theta) \cdot p(\theta)}{p(\mathbf{w})} \tag{8}$$

where θ is the random variable for spam, $\mathbf{w}$ is a bag of words, $p(\theta|\mathbf{w})$ is the posterior distribution, $p(\mathbf{w}|\theta)$ is the likelihood, $p(\theta)$ is the prior distribution and $p(\mathbf{w})$ is a normalisation constant, also called evidence, which is irrelevant for classification. Now, assuming that each feature is independent of the others it holds that

$$p(\theta|\mathbf{w}) = \prod_{i=1}^{m} p(w_i|\theta) \tag{9}$$

where $p(w_i|\theta)$ are the class-conditional probabilities (=probability of observing an email with word w_i that belongs to spam). Since $p(\mathbf{w})$ is only a normalisation constant, the formula can be further simplified to stating

$$p(\theta|\mathbf{w}) \propto p(\mathbf{w}|\theta)p(\theta) \tag{10}$$

An email is then classified as spam, if its probability to be spam $p(Spam|\mathbf{w})$ is higher than its probability to be ham $p(Ham|\mathbf{w})$. In some instances the Naive Bayes classifier can be quite inaccurate, mainly due to the lack of a subjective prior probability and the independence assumption when combining the single feature probabilities. We therefore applied some other approaches and compared them to Naive Bayes.

7.2 Derivation of the posterior distribution

The Beta prior is defined by

$$p(\theta) = B(\theta|\alpha_s, \alpha_h) = \frac{\Gamma(\alpha_s + \alpha_h)}{\Gamma(\alpha_s)\Gamma(\alpha_h)} \theta^{\alpha_s - 1}(1-\theta)^{\alpha_h - 1} \tag{11}$$

where $\Gamma(\cdot)$ is the Gamma distribution and $\alpha_s > 0$ and $\alpha_h > 0$ are the hyperparameters. Since our outcome is binomial distributed (spam or ham), the likelihood of a

word being a spam- or a ham-word is given by

$$p(w|\theta) = \theta^{n_s}(1-\theta)^{n_h} \tag{12}$$

As mentioned in section 7.1, it holds that the posterior is proportional to the likelihood times the prior, i.e. $p(\theta|w) \propto p(w|\theta) \cdot p(\theta)$. Thus,

$$\begin{aligned} p(\theta|w) &\propto p(w|\theta) \cdot p(\theta) && (13)\\ &\propto \theta^{n_s}(1-\theta)^{n_h} \frac{\Gamma(\alpha_s+\alpha_h)}{\Gamma(\alpha_s)\Gamma(\alpha_h)} \theta^{\alpha_s-1}(1-\theta)^{\alpha_h-1} && (14)\\ &\propto \frac{\Gamma(\alpha_s+\alpha_h)}{\Gamma(\alpha_s)\Gamma(\alpha_h)} \theta^{n_s}\theta^{\alpha_s-1}(1-\theta)^{n_h}(1-\theta)^{\alpha_h-1} && (15)\\ &\propto \frac{\Gamma(\alpha_s+\alpha_h+N)}{\Gamma(\alpha_s+n_s)\Gamma(\alpha_h+n_h)} \theta^{\alpha_s+n_s-1}(1-\theta)^{\alpha_h+n_h-1} && (16) \end{aligned}$$

7.3 Combining the Feature Probabilities

Other methods for combining the probabilities, described in Winkler (2016), are the Stouffer and Mudholkar–George methods.
The Stouffer method is based on Z-scores and thus depends on the standard normal distribution. It combines probabilities in the following way

$$H = \Phi^{-1}\left(\frac{1}{\sqrt{(M)}} \prod_{i=1}^{M} (1 - f(w_i))\right) \tag{17}$$

$$S = \Phi^{-1}\left(\frac{1}{\sqrt{(M)}} \prod_{i=1}^{M} (f(w_i))\right) \tag{18}$$

Another possible way to combine the feature probabilities is the Mudholkar–George method. It uses the inverse of the t distribution with $5M+4$ degrees of freedom and is defined as follows

$$H = t^{-1}\left(\frac{1}{\pi}\sqrt{\frac{3(5M+4)}{M(5M+2)}} \ln \prod_{i=1}^{M}\left(\frac{1-f(w_i)}{f(w_i)}\right), 5M+4\right) \tag{19}$$

$$S = t^{-1}\left(\frac{1}{\pi}\sqrt{\frac{3(5M+4)}{M(5M+2)}} \ln \prod_{i=1}^{M}\left(\frac{f(w_i)}{1-f(w_i)}\right), 5M+4\right) \tag{20}$$

These three methods (the third one being the Fisher method explained in Section 7.3) with different underlying distributions lead to different rejection regions and thus influence the final classification.

7.4 PCC and F1

The intuiton behind PCC and F1 as accuracy measures is given as follows. Precision (PCC / p) answers "Out of all the examples the classifier labeled as positive, what fraction was correct?" On the other hand, recall (r) answers "Out of all the positive examples there were, what fraction did the classifier pick up?" Since the F1 metric considers *precision* as well as *recall*, a good performance on both is favoured over a high performance on one and a poor performance on the other. Like already mentioned in Section 4, it is more important that ham emails are correctly retrieved and specified and thus we would like to maximize the F1 score for spam emails because it strongly depends on the *false negatives* of the ham emails.

7.5 Results

Method	s_{opt}	x_{opt}	PCC_{Spam}	F1_{Spam}	PCC_{Ham}	F1_{Ham}
Fisher	11	0.20	93.68%	95.70%	99.04%	98.09%
Stouffer	11	0.38	72.28%	63.38%	74.48%	79.59%
Mudholkar	10	0.18	93.68%	95.36%	98.72%	97.93%

Table 1: Hyperparameters and PCC for three different combinations methods

Method	s_{opt}	x_{opt}	F1_{Spam}	p_{Spam}	r_{Spam}	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 2: Hyperparameters, F1 score and its components for different combination methods

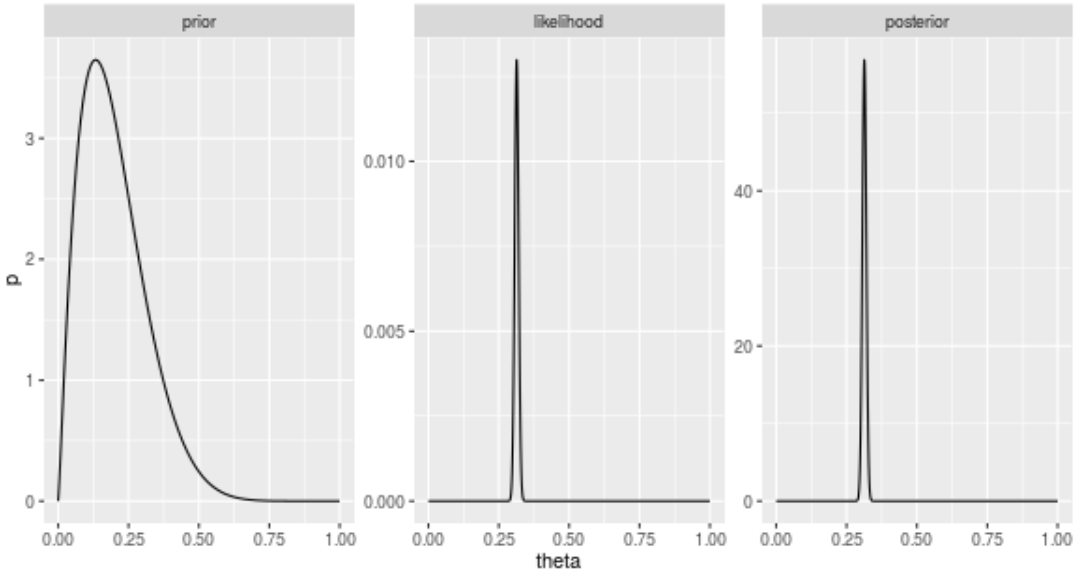

Figure 1: Prior, likelihood and posterior distribution for PCC

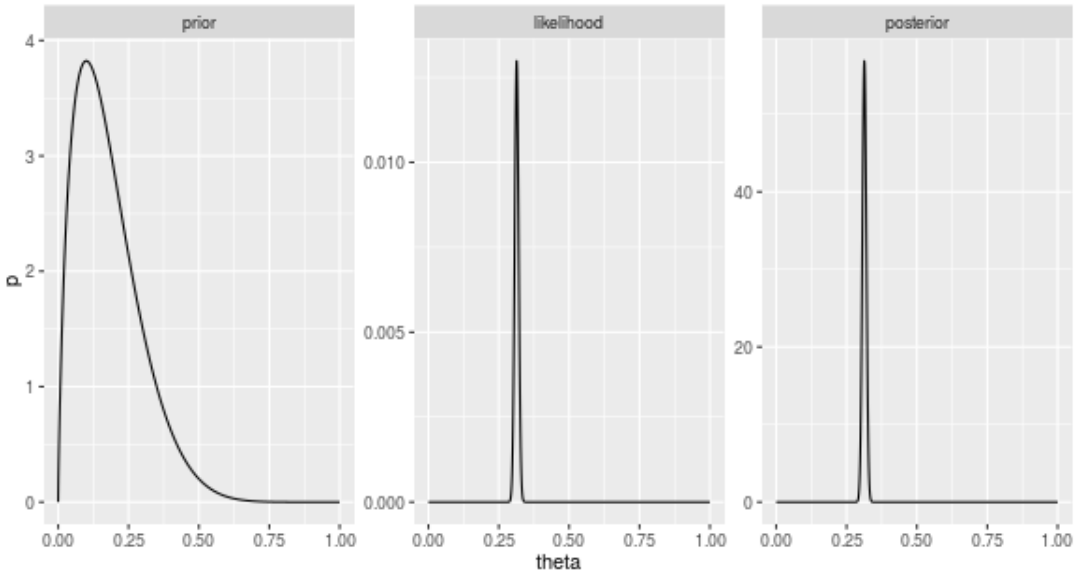

Figure 2: Prior, likelihood and posterior distribution for F1 score