Evaluation Of Linear Interpolation Smoothing On Naive Bayes Spam Classifier

Adewole A.P, Fakorede O.J, Akwuegbo S.O.N

(University of Lagos) Department of Computer Sciences, University of Lagos, Lagos, Nigeria; [2](Federal University of Agriculture, Abeokuta) Department of Statistics, Federal University of Agriculture, Abeokuta, Nigeria.
Email: padewole@unilag.edu.ng; josephkorede@yahoo.com, agwuegbo_son@yahoo.com

ABSTRACT: The inconvenience associated with spams and the cost of having an important mail misclassified as spam have made all efforts at improving spam filtering worthwhile. The Naive Bayes algorithm has been found to be successful in properly classifying mails. However, they are not perfect. Recent researches have introduced the idea of smoothing into the Naive Bayes algorithm and they have been found to produce better classification. This study applies the concept of linear interpolation smoothing to Naive Bayes spam classification. The resulting classifier did well at improving spam classification and also reducing false positives.

Keywords: Naïve Bayes, Smoothing, Linear Interpolation, Spam, Ham False Positives, False Negatives.

The inexpensive cost of sending emails has come with the undesirable activity of sending unwanted emails by largely unknown sources. These unwanted emails are known as spams. To a sender of a spam, sending it is quite cheap. However, to the receiver a spam comes in at an expense in terms of money, time and discomfort. Today, it is estimated that spam costs organizations globally billions of dollars in terms of lost productivity and equipment, software and manpower needed to combat the problem. Also, spams contribute to other crimes, such as: financial theft, identity theft, data and intellectual property theft, and virus and malware infection [5]. Spams are also used in marketing pornography and other objectionable materials. The aforementioned implications of spam have made efforts in fighting or even eliminating spams worthwhile. Several methods as highlighted by Metz [6] have been used overtime in combating spams with varying degree of success. Of all the methods used in combating spam, the Naive Bayes algorithm has been found to be largely successful. The Naive Bayes classifier combines the probabilities of every token in a mail, and estimates the probability of the mail being spam using Bayes' theorem. It should be noted that it is possible that in a given mail to be classified that certain tokens that were not seen during training are present [20]. The probabilities of the unseen tokens if calculated using the traditional maximum likelihood estimate yields zero which can consequently reduce the combine probabilities to zero [4], [11],[14],[20]. This problem can be handled using the language model approach in a process known as smoothing**.** Smoothing basically increases the probability of unobserved items at the expense of the probability of observed items. In other words, smoothing describes techniques for adjusting the maximum likelihood estimate to hopefully produce more accurate probabilities [15], [17]. A Bayesian classifier for spam filtering with more accurate probability can produce a better classification and ultimately reduce *false positives* which are the aim of most spam filters. There are several methods of smoothing; some of which are Laplace, Witten-Bell, Lidstone, Katz, Two-stage smoothing, Kneser-Ney, Church-Gale, Absolute Discounting, Linear interpolation (also known as Jelinik-Mercer Smoothing) among others [14],[17]. The aim of this study basically is to evaluate Linear Interpolation smoothing albeit a modified approach (an interpolation of Laplace estimate and the maximum likelihood estimate of the collection) on Naive Bayes spam classifiers. It is believed that well smoothed probabilities can have a positive influence on the performance of a classifier in terms of accuracy and in spam filtering in terms of reducing *false positives.*

2 NAIVES BAYES SPAM CLASSIFIER AND SMOOTHING

A Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem with naive independence assumptions.

2.1 Bayes Probabilistic Model

Bayes probability model for a classifier is a conditional model *P(C|*$x_1, \ldots, x_n$*)* over a dependent class variable *C* with a small number of outcomes or *classes*, conditional on several feature variables $x_1$through x_n [17]. The Bayes probability model is given as:

$$P(c|x) = P(c) * P(x|c)/P(x) \quad (1)$$

In plain English the above equation can be written as:

$$Posterior = prior * likelihood/evidence \quad (2)$$

2.2 Naïve Assumption Bayes Model

Given a set of features { $x_1, \ldots, x_n$} over a set of class{$c_1, \ldots, c_m$}, then if the number of features *n* is large, calculating$P(x|c)$ becomes difficult,time consuming and may be infeasible. In order to simplify the calculation of $P(x|c)$, an assumption of independence of features is made. This independence assumption of features led to the name *Naive Bayes.* If the assumption of independence is made, then$P(x|c)$ is calculated as:

$$P(x|c) = \prod_{i=1}^{n} P(x_i|c) \quad (3)$$

If (3) is substituted into (1) then, we have,

$$P(C|x) = P(C) * \prod_{i=1}^{n} P(x_i|C)/P(x) \quad (4)$$

2.3 Spam Classifier

Given a set *M = {S, L},* then a spam filter can be seen as a decision function that decides if an incoming mail M is spam*(S)* or legitimate *(L).*If the set of all e-mail messages is denoted by M, then the spam filter *f* is given by *f : M* →

{S,L}. This function is obtained by training a machine learning algorithm on a set of pre-classified *messages* $\{(m_1,c_1), (m_2,c_2). . . (m_n, c_n)\}, m_i \in M, c_i \in \{S,L\}$ [12].

2.4 How A Naive Bayes Spam Classifier Is Built

The naive Bayes spam classifier has two parts namely: training part and the classification part. The training aspect involves the following steps:

- **I.** *Tokenization:* This step is very important as this allows the set of tokens constituting a training corpus to be obtained.
- **II.** *Representation:* Most machine learning algorithms only classify numerical objects (real numbers or vectors) [12]. The set of tokens obtained in the previous step must be represented in some ways. Notable methods of representation include the bag-of-words representation, Term frequency-inverse document frequency (Td-idf), binary representation among others [18].
- **III.** *Feature Selection:* After a mail is tokenized, the resultant tokens obtained are usually very large. The number of tokens that may result at this stage may run into tens or even hundreds of thousands. This has the disadvantage of causing high computational complexity. In order to reduce the dimension of the feature space, some feature selection methods are used to remove less informative terms. Popular feature selection methods include: Information Gain, Document Frequency, Term Frequency Variance, Chi-square method and Mutual Information.
- **IV.** *Probability Estimation*: From the Bayesian probability model given in equation (i), *P(x|C)* corresponds to the likelihood of seeing a feature x in class C. With respect to Spam and Non Spam classes, we can rewrite P(x|C) for a token ***t*** as:

P(t|Spam)=$frequency\ of\ t\ in\ Spam/total\ n\ tokens\ in\ Spam$ (5)

P(t|Ham) = $frequency\ of\ t\ in\ Ham/total\ n\ tokens\ in\ Ham$ (6)

P(C) corresponds to the prior probabilities of the classes of interest.

P(Spam)=
$number\ of\ Spam\ messages/Total\ number\ of\ messages$ (7)

P(Ham)=
$number\ of\ Ham\ messages/Total\ number\ of\ messages$ (8)

P(x) as it is seen in equation (1) is usually not calculated as it is the same for all classes.

- **V.** The probabilities computed in step (IV) are used in calculating *P(C|x)*

The classification aspect consists of the following steps:

- **I.** The incoming mail is tokenized.
- **II.** Obtain the corresponding likelihood estimate for each of the resulting tokens.
- **III.** Combine the likelihood estimates for all the tokens found in the mail
- **IV.** Compare the combined estimate for each class and return the class with higher value as the class of the inspected mail i.e. *maximum a posteriori* or *MAP* decision rule.

2.5 Smoothing

A situation may arise when a feature or a token does not appear at all during training. In such a situation, the probability of unseen tokens becomes zero. This obviously cannot be used as input to the Bayesian classifier as it will reduce everything to zero. Smoothing can be used in handling cases like this. Smoothing aims to adjust the probability of an unseen event from a seen event, that arise due to data sparseness. Some popular smoothing methods include: Laplace Smoothing, Good-Turing Estimate, Witten-Bell Estimate, Jelinek-Mercer Smoothing(Linear Interpolation), Absolute Discounting, Dirichlet Smoothing, Two stage smoothing e.tc.

2.5.1 Benefit of Smoothing

As earlier stated, the major motivation for smoothing is to handle zero frequency problems that may occur in document models. It has however been observed overtime that smoothing can improve information retrieval and Bayesian classification. Zhai and Lafferty [2] theorized that smoothing play dual roles. These roles are:

- It improves the reliability of the model, especially by assigning non-zero probabilities to terms that do not occur in the document.
- It facilitates the generation of terms in the query that are commonly used in general or are particularly typical in the collection.

3 METHODOLOGY

3.1 Tokenization

The corpus used for training and testing the classifier were tokenized using whitespaces and punctuation marks such as “?”,”.” and ”!”. Numerical values were also dropped as they will not add to classification efficiency. Dropping numerical values also allow whatever character that may be appended at the front of the number ignored to be retained. For example, if we have a string like *“$4000”,* by ignoring the number, the character “$” can be retained as a token. This becomes particularly useful in spam filtering because characters representing currencies especially dollars occur frequently in spams.

3.2 Feature Selection

None of the feature selection methods enumerated in section 2.4 was used. However, some steps were taken to reduce the dimension of the set of tokens and also ensure that words that were not considered to be important were screened out. This was done by removing words containing less than four characters.

3.3 Probability Estimation

Prior Probability: The prior probabilities for the two categories under consideration i.e spam and ham were computed as:

P (spam) = $number of spam\ mails/total\ number\ of\ mails$

P (ham) = $number of ham\ mails/total\ number\ of\ mails$

Likelihood: The likelihood was calculated using the interpolation of the Laplace estimate of a token and the maximum likelihood estimate of that token in the total corpus collection. In other words, a parameter λ was introduced such that:

Given a mail $M = \{S, L\}$, the likelihood estimated of a token *t* in category *S* is computed as

$$P_{\lambda,t}(t|S) = (1-\lambda)P_{LAP}(t|S) + \lambda P_{MLE}(t|M) \quad (4)$$

where $P_{LAP}(t|S)$ is the Laplace estimate of a token *t* in category *S* and $P_{MLE}(t|M)$ is the maximum likelihood estimate of token t in the total collection of *S* and *L* i.e *M* Equation (4) was adopted from *Jelinek-Mercer's* linear interpolation smoothing [7] albeit with the introduction of Laplace smoothing in it. The parameter λ is known as the smoothing parameter.

3.4 Naïve Bayes Model Used

Multinomial Naive Bayes model was used in the implementation. The major reason why it is preferred is because Multinomial model performs well as compared to other models such as Bernoulli and Binarized Multinomial Models [1].

3.4 Performance Evaluation

The performance of the constructed filter was evaluated from two different stand points. These are decision theory (false positives and false negatives) and information retrieval (recall and precision) With respect to spam classification, false positives are legitimate messages classified as spam while false negatives are spam messages classified as legitimate. *Recall* is the ratio of relevant items that are retrieved, which in this case is the proportion of spam messages that are actually recognized. For example if 9 out of 10 spam messages are correctly identified as spam, the recall rate is 0.9. *Precision* is the ratio of the spam messages classified as spam over the total number of spam messages tested.

4 EXPERIMENTS AND RESULTS.

The main purpose of the project is to test how the interpolation of Laplace estimate and the maximum likelihood estimate of each token in the total corpus will fare. To test this, the raw *LingSpam* corpus ((i.e the corpus was not lemmatized or pre-processed in any way) was used to train the filter. The corpus used for this project is the *LingSpam Corpus* obtained from *http://csmining.org/index.php/ling-spam-datasets.html* [13]. A portion of the corpus was also reserved for testing the performance of the filter. Overall, over 150 mails were used to train each category of interest (i.e Spam and Ham classes). Recall that the formula used for calculating the likelihood of each of the features in each category is:

$$P_{\lambda,t}(t|S) = (1-\lambda)P_{LAP}(t|S) + \lambda P_{MLE}(t|M)$$

The likelihood estimate given above allows a feature or a token to be seen from two perspectives: its class and in all the classes, where the parameter λ serves as an adjustment parameter. To test the spam filter, 50 mails of spam and legitimate classes were each used for this experiment. Generally, the spam filter works well however, the following observations were made:

I. When λ >= 0. 6, the spam filter performed relatively poorly.

II. When λ = 0.9, misclassifications were very high.

III. Apart from when λ = 0.1, there were at least 2 misclassifications for the values of λ between 0.1 and 1.

IV. Generally, there were more spams misclassified as legitimate mails as legitimate mails misclassified as spam.

The table below shows the performance of the filter built

λ	TN	TP	FN	FP	PRE (%)	REC (%)	ACC (%)
0.9	40	20	30	10	66.6	40	60
0.5	46	40	10	4	90.9	80	86
0.1	49	47	3	1	97.9	94	96

Figure 1

Note:
TN refers to legitimate mail classified as legitimate.
TP refers to spam classified as spam.
FP refers to legitimate mail classified as spam
FN refers to spam classified as legitimate mail.
PRE implies precision
REC refers to the recall
ACC refers to accuracy.

Precision = $\boldsymbol{TP}/(\boldsymbol{TP}+\boldsymbol{FP})$
Recall = $\boldsymbol{TP}/(\boldsymbol{TP}+\boldsymbol{FN})$
Accuracy = $(\boldsymbol{TP}+\boldsymbol{TN})/(\boldsymbol{TP}+\boldsymbol{TN}+\boldsymbol{FP}+\boldsymbol{FN})$

From the above table, it can be inferred that the filter performs best when λ is tuned to 0.1. An astonishing observation made in the table above is that at λ= 0.9, when the TP is low the TN is high. This is likely due to the nature of the corpus used i.e the *Lingspam* corpus. The tokens making up the spam corpus falls in a wider area of discourse compared with those making up the ham.

4 CONCLUSION

The major aim of this paper is to see how the interpolation of Laplace and the maximum likelihood estimates of the collection model can improve the performance of spam filters and also reduce the number of false positives. It has been seen that when the smoothing parameter is rightly set, the classification can be very efficient and also that the number of false positives can be reduced.

REFERENCES

[1] K. McCallum, Nigam, "A comparison of event models for naive Bayes text classification". AAAI/ICML- 98Workshop on Learning for Text Categorization, AAAI Press 41–48, 1998.

[2] C. Zhai, and J. Lafferty, "The Dual Role of Smoothing in the Language Modelling Approach". In Proceedings of the Workshop on Language Models for Information Retrieval (LMIR) 2001, pages 31–36, 2001.

[3] C. Zhai and J. Lafferty, "A Study of Smoothing Methods for Language Models Applied to ad hoc Information Retrieval", 2001.

[4] D. Vilar, H. Ney, A. Juan, and E. Vidal, "Effect of Feature

Smoothing Methods in Text Classification Tasks". In International Workshop on Pattern Recognition in Information Systems, pages 108-117. Porto, Portugal, 2004.

[5] D. Anderson "Statistical Spam Filtering", http://www.web.eecs.umich.edu/rthomaso/courses/nlp2006/David_Anderson.pdf , 2006.

[6] D. Metz , "International Business Machines(IBM) accessed 27th February 2014 <www.ibm.com/developerworks/linux/library/l-spamf/index.html>

[7] F. Jelinek, and R.L. Mercer, "Interpolated estimation of Markov source parameters from sparse data", in Proc. Workshop on Pattern Recognition in Practice, pages 381-397, Amsterdam, 1980L.

[8] H.C. Hong (STEVEN), "Statistical Machine Learning for Data Mining and Collaborative Multimedia Retrieval", The Chinese University of Hong Kong, 2006.

[9] I. Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C. Spyropoulos, and P. Stamatopoulos. Learning to filter spam email: A comparison of a naive bayesian and a memory based approach. Workshop on Machine Learning and Textual Information Access, 4, 2000.

[10] I. Androutsopolous, G. Paliouras, E. Michelakis, "Learning to Filter Unsolicited Commercial E-Mail". Athens University of Economics and Business and National Centre for Scientific Research "Demokritos", 2004.

[11] Jon Kagstrom "Improving Naive Bayesian Spam Filtering", Mid Sweden University, Sweden, 2005.

[12] K. Tretyakov, *Machine Learning Techniques in Spam Filtering,* Institute of Computer Science, University of Tartu, Estonia. pp 3-5, 7-8, 2004.

[13] Lingspam Corpus [Online], Available: *http://csmining.org/index.php/ling-spam-datasets.html*

[14] N. A. Abdulmutalib, "Language Models and Smoothing Methods for Information Retrieval", Ph.D dissertation, Department of Computer Science, University of Dortmund Dortmund, Germany , 2010.

[15] Q. Yuan, G. Cong, and N.M. Thalmann, " Enhancing Naive Bayes with Various Smoothing Methods for Short Text Classification", in Proc. of the 21st international conference companion on World Wide Web, WWW "12 Companion, 2012.

[16] M. Sahami, S. Dumais, D. Heckerman, and E. Harvitz, "A Bayesian approach to filtering Pg 2-4, 2008.

[17] S. F. Chen and J. Goodman, "An Empirical Study of Smoothing Techniques for Language Modeling", Computer Science Group Harvard University Cambridge, Massachusetts,1998.

[18] S.T. Guzella and W.M. Caminhas, "A review of machine learning approaches to filtering", Department of Electrical Engineering, Federal University of Minas Gerais, Brazil,2009 (www.elsevier.com/locate/eswa).

[19] Wikipedia, "Naïve Bayes Classifier", http://en.wikipedia.org/wiki/naive Bayes classifier, 2014.

[20] X. Zhou, X. Zhang, X. Hu, "Semantic Smoothing for Bayesian Text Classification with Small Training Data", College of Information Science & Technology, Drexel University, Philadelphia, 2008.

[21] Y. Yang, and J.O. Pederson, "A comparative study on feature selection in text categorization". In Fisher, D.H., ed.: Proceedings of ICML-97, 14th International Conference on Machine Learning, Nashville, US, Morgan Kaufmann Publishers, San Francisco, 412–420, 1997.