Research on Spam Filters Based on NB Algorithm

Shengyue Su

DUT—RU International School of Information Science & Engineering, Dalian University of Technology, 116000 Dalian, China

Abstract. Spam filtering is a crucial part of network security. As spam becomes more complex, traditional rule-based methods struggle to meet the needs of modern email systems. The SpamAssassin dataset is used in this study to explore the use of the Naive Bayes (NB) algorithm for spam detection. The algorithm demonstrated high accuracy and efficiency in classifying large-scale text data, achieving an accuracy of 97.74%, a recall rate of 96.60%, and a precision rate of 96.8%, with an F1 score of 0.97. Through confusion matrix and Receiver Operating Characteristic (ROC) curve analyses, the model's effectiveness in spam filtering was demonstrated by its high True Positive Rate (TPR) and low False Positive Rate (FPR). However, limitations arise from the NB algorithm's independence assumption, which may affect performance in more complex spam scenarios. Future work may focus on improving the model's accuracy and robustness by integrating it with other machine learning models, like Support Vector Machines (SVMs) and deep learning techniques, to enhance spam classification capabilities.

1 Introduction

Due to the rapid advancement of the internet, communication on a daily basis cannot survive without email, but the issue of spam has become even more serious. Spam is not just wasting time and bandwidth resources for users, it can also spread malware and phishing attacks, putting their privacy and security at risk [1]. Traditional rule-based spam filtering methods rely on manually set rules, such as blacklists and keyword matching, which are becoming increasingly inadequate in dealing with modern, complex spam forms [2].

In recent years, machine learning technology, like those described in Gupta et al. [3], especially the NB algorithm, have become one of the mainstream methods for spam filtering due to their simplicity, ease of use, and computational efficiency. However, the independence assumption of NB poses limitations in spam detection [2, 4]. To address these issues, researchers have proposed various improvements, like combining NB with SVMs and deep learning techniques to enhance the classifier's robustness and accuracy [1, 4, 5].

Corresponding author: 1049190808@mail.dlut.edu.cn

2 Related work

The evolution of spam filtering technology has transitioned from rule-based methods to machine learning models. Early spam filters relied on manually written rules and keyword matching [2]. However, as spam became more complex, traditional methods gradually lost their effectiveness. The NB algorithm was one of the earliest machine learning methods used for spam detection, and since its proposal for this purpose in 1998, it has been widely applied due to its efficiency and ease of implementation [6, 7]. As Tretyakov discusses in his comprehensive review, the simplicity and efficiency it provides in handling high-dimensional feature spaces is what makes the NB classifier so popular in spam filtering. [8]. However, the independent assumption of the NB algorithm causes performance degradation when dealing with complex data [2].

To overcome this issue, researchers have developed several improved algorithms, such as hybrid models that combine NB with SVMs, which have shown excellent performance when processing high-dimensional data [5, 9]. Tretyakov's work provides an in-depth analysis of SVM's application in spam filtering, particularly highlighting its ability to find optimal separating hyperplanes in feature space [8]. In addition, recent deep learning techniques, like Long Short-Term Memory networks and Convolutional Neural Networks, have been utilized in the realm of spam filtering, significantly improving detection accuracy, as highlighted in Gupta et al. [3].

3 Methodology

3.1 NB overview

Bayes' Theorem is used to build the probabilistic classification algorithm called NB Classifier. Its simplicity and computational efficiency have resulted in its widespread usage in areas like text classification and spam filtering. And the basic form of Bayes' Theorem is:

$$P(A|B) = \frac{P(B|A)\cdot P(B)}{P(A)} \quad (1)$$

If event B has already happened, the probability of event A happening is $P(A|B)$. Features are thought to be independent by the NB classifier, meaning that each feature's contribution to the classification result is unaffected by other features. Despite this assumption not holding true in many real-world scenarios, NB classifiers are still effective in a variety of applications, particularly when it comes to text classification tasks.

3.2 Bayesian algorithm in spam filtering

In the application of spam filtering, the NB algorithm calculates the conditional probability of each word in an email being associated with spam or non-spam to determine the email's category. The algorithm assumes that the appearance of words is independent and classifies emails by calculating how frequently words appear in non-spam and spam emails. The classification process can be expressed as:

$$P(\text{spam}|\text{message}) = \frac{P(\text{message}|\text{spam})\cdot P(\text{spam})}{P(\text{message})} \quad (2)$$

The likelihood that a given email is spam is $P(\text{spam}|\text{message})$. The likelihood that the specific message content will be included in spam emails is $P(\text{message}|\text{spam})$. The likelihood that an email is spam is $P(\text{spam})$. The likelihood that the message content will be included in all emails is $P(\text{message})$.

The probability that an email is spam is calculated by the NB classifier by analyzing the frequency of each word in its “bag of words” approach. For example, words like "Free" or "Viagra" are more susceptible to appearing in spam emails than in non-spam emails, which increases the likelihood that the email will receive spam labeling [6].

3.3 Data preprocessing

Before applying the NB classifier, preprocessing the raw email data is crucial. Preprocessing typically consists of:

• Text cleaning involves removing unnecessary characters, like HTML tags and punctuation.

• Tokenization splits the email content into individual words.

• Stop word removal eliminates high-frequency words, such as "the" or "is," that do not significantly contribute to classification [10].

• Word frequency calculation converts the frequency of each word into a vector, building a model called "bag of words".

Through this series of preprocessing steps, the email content is transformed into feature vectors suitable for the NB classifier to handle.

3.4 Feature selection

Selecting features is an important aspect of optimizing model performance, as it aims to pinpoint the most crucial features for classification. By selecting key features, the model's computational complexity can be reduced while improving its accuracy. Common feature selection methods include information gain and chi-square testing.

3.5 Classifier evaluation

In spam filtering tasks, it is crucial to evaluate the classifier's performance. Evaluation metrics that are commonly used include F1 Score, recall, precision, and accuracy. By utilizing these metrics, a comprehensive evaluation of the classifier's spam detection performance can be made, allowing for an assessment of its practical application.

4 Experimental results

This experiment is designed to evaluate the effectiveness of the NB classifier in spam classification tasks. The dataset used is the SpamAssassin email dataset, which includes two categories: normal (Ham) and spam (Spam) emails. The data distribution is as follows:

- Easy Ham 1 consists of 2,500 normal emails.
- Easy Ham 2 consists of 1,400 normal emails.
- Spam 1 consists of 500 spam emails.
- Spam 2 consists of 1,400 spam emails.

These subsets are used for training and testing the NB model, assessing its performance across varying data volumes and category distributions. The dataset ensures an appropriate ratio of normal and spam emails in the training set for accurate evaluation on the test set.

4.1 Data preprocessing

To ensure the email data could be effectively processed by the classifier, the raw email data underwent the following preprocessing steps:

•HTML tag cleaning: Since many emails are formatted in HTML, these tags were removed to extract pure text.

•Tokenization and feature extraction: Each email was split into individual words, then a vocabulary was built, and the emails were converted into vector form using the model named "bag of words". This step quantified the input features for the classifier.

•Label generation: All normal emails were labeled as "non-spam" (Ham), while all spam emails were labeled as "spam" (Spam) for training and evaluation purposes.

4.2 Experimental results

After running experiments on the SpamAssassin dataset, the spam classification task's performance was satisfactory for the NB classifier as per the following:

•Accuracy: The classifier was able to correctly categorize most emails, regardless of whether they were spam or non-spam, with an overall accuracy of 97.74%.

•Recall: The recall rate of the classifier was 96.60%, indicating that the classifier was able to identify a majority of spam emails, with a few false negatives.

•Precision: The precision rate of the classifier was 96.8%, implying that the majority of emails that were flagged as spam actually were.

•F1 Score: The F1 score of 0.97 showed an excellent mix between recall and precision.

4.3 Confusion matrix analysis

The model's confusion matrix is displayed in Fig. 1, which visually reflects the classification performance of the classifier.

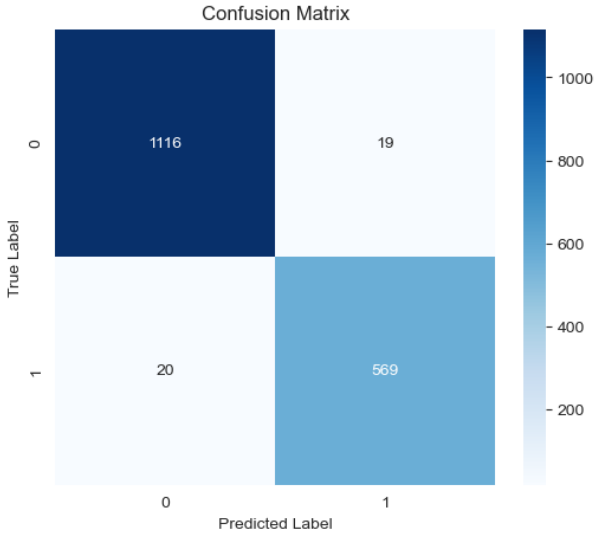

Fig. 1. Confusion Matrix (Picture credit : Original)

- True Negatives : 1,116 emails that were not spam were correctly identified as non-spam.
- False Positives : 19 emails that were not spam were mistakenly identified as spam.

• False Negatives : 20 emails that were spam were mistakenly identified as non-spam.
• True Positives : 569 emails that were spam were correctly identified as spam.

These results indicate that the classifier accurately identifies non-spam emails, with a low false positive rate. Additionally, the model performs well in detecting spam, although a small number of spam emails were missed.

4.4 ROC Curve Analysis

Fig. 2 shows the ROC curve, which displays how the classifier performs at different threshold settings.

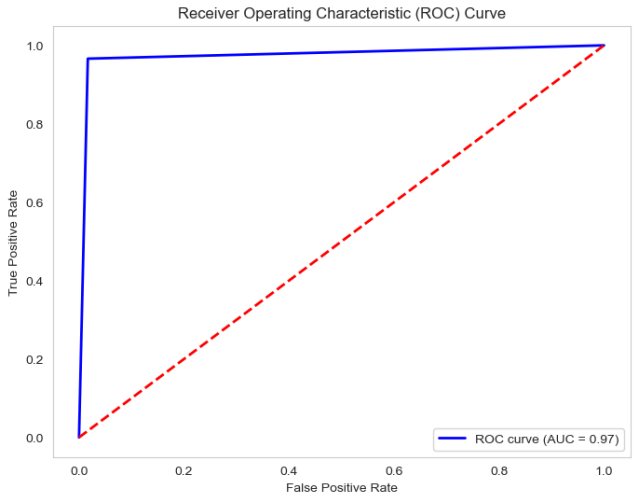

Fig. 2. ROC Curve (Picture credit : Original)

• The classifier's Area Under the Curve (AUC) of 0.97 attests to its high level of discrimination between non-spam and spam emails.
•TPR approaches 1, meaning the model successfully identifies most spam emails.
•FPR is low, showing that the model rarely misclassifies normal emails as spam.

4.5 Discussion

It is clear from the experiment findings that the NB classifier provides the following benefits:

• High precision and accuracy: With a low error rate and effective distinction between non-spam and spam emails, the model is suitable for large-scale email filtering tasks.
• Good F1 score and recall: The model captures most spam emails while maintaining high precision, leading to a high F1 score.
• Stability reflected in the ROC curve: The AUC value of 0.97 shows that the classifier performs consistently across various thresholds, proving its reliability in spam filtering.

Although the classifier performs well, there may still be cases where emails that are not spam are misclassified as spam, and some emails that are spam may go undetected. In the future, integrating other models, like SVMs or deep learning models, may further improve classification accuracy and decrease false positives and negatives. Moreover, testing the classifier on more diverse datasets is a potential research direction.

5 Conclusion and Future Outlook

The NB algorithm, as an efficient and straightforward method for spam filtering, has been widely used for decades. However, as spam types diversify and become more complex, the NB algorithm alone has certain limitations in dealing with modern spam. Future research may focus on further optimizing the NB algorithm or combining it with other machine learning algorithms to enhance the robustness and accuracy of detection.

Simultaneously, researchers can further investigate the use of deep learning models in spam detection, especially in handling large unstructured data. Additionally, with the proliferation of IoT devices, how to efficiently run spam filtering algorithms on resource-constrained devices is another key research area. By optimizing model structures and introducing stronger feature selection techniques, NB and its improved methods still have broad application prospects in spam detection.

References

1. A. Makkar, S. Garg, N. Kumar, M. S. Hossain, A. Ghoneim and M. Alrashoud, An Efficient Spam Detection Technique for IoT Devices Using Machine Learning. IEEE Transactions on Industrial Informatics. **17**, 903-912 (2021)
2. T. Wu, S. Liu, J. Zhang, Y. Xiang, Twitter spam detection based on deep learning, in Proceedings of the ACSW '17: Proceedings of the Australasian Computer Science Week Multiconference, Geelong, Australia, Association for Computing Machinery, (2017), 1-8
3. S. D. Gupta, S. Saha and S. K. Das, SMS spam detection using machine learning. J. Phys. Conf. Seri. **1797**, 012017 (2021).
4. P. Navaney, G. Dubey and A. Rana, SMS Spam Filtering Using Supervised Machine Learning Algorithms, in Proceedings of the 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, IEEE, (2018), 43-48
5. W. Feng, J. Sun, L. Zhang, C. Cao and Q. Yang, A support vector machine based NB algorithm for spam filtering, in Proceedings of the 2016 IEEE 35th International Performance Computing and Communications Conference (IPCCC), Las Vegas, NV, USA, IEEE, (2017), 1-8
6. N. Kumar, S. Sonowal and Nishant, Email Spam Detection Using Machine Learning Algorithms, in Proceedings of the 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, IEEE, (2020), 108-113
7. W. A. Awad and S. M. ELseuof, Machine learning methods for spam e-mail classification. International Journal of Computer Science & Information Technology (IJCSIT). **3**, 173-184 (2011)
8. K. Tretyakov, Machine learning techniques in spam filtering, in Proceedings of the Data mining problem-oriented seminar, MTAT, (2014), 60-79
9. D. S. Jawale, A. G. Mahajan, K. R. Shinkar and V. V. Katdare, Hybrid spam detection using machine learning. International Journal of Advance Research, Ideas and Innovations in Technology. **4**, 2828-2832 (2018)
10. N. F. Rusland1, N. Wahid1, S. Kasim1 and H. Hafit1, Analysis of Naïve Bayes algorithm for email spam filtering across multiple datasets. IOP Conf. Ser.: Mater. Sci. Eng. **226**, 012091 (2017)