The effect of using Naive Bayes to detect spam email

Zehui Sun

Department of EIE, The Hong Kong Polytechnic University, Hong Kong, China

Abstract. The rapid growth of the Internet has made email a huge help in business, life, and education. However, it has also become a channel for spreading undesirable information, such as content from hackers, viruses, violence, pornography, and superstition. Spam, which is unsolicited commercial email, often carries such undesirable information. It wastes network bandwidth, consumes users' precious time, and interferes with normal life. Therefore, spam detection and filtering have become especially urgent and of great practical importance. This paper focuses on the spam detection method based on the plain Bayesian algorithm. The plain Bayesian algorithm is particularly suitable for spam detection due to its high detection accuracy and its wide application in text classification tasks. The results and analysis of the experimental dataset demonstrate that the accuracy of Park's Bayesian algorithm in spam detection reaches an impressive 99.193%. This high level of accuracy underscores the effectiveness of the Bayesian approach in identifying and filtering out spam, thereby enhancing the overall efficiency and security of email communication.

1 Introduction

Email's rapid expansion on the Internet has made people's lives, careers, and intellectual endeavors much easier, but it has also provided a channel for the spread of bad information such as hackers, viruses, reactions, violence, pornography, and superstition. Spam is an unsolicited commercial email that usually carries bad information, wasting limited network bandwidth and users' precious time, and disrupting people's normal lives. In addition, when receiving a large amount of spam emails, users are more likely to ignore important non-spam emails, which means many email users must spend time cleaning these spam emails. With time, the detection and filtering of email spam has become urgent and a research topic with global and important practical significance [1-4].

The main email detection algorithms based on statistical methods include the Naive Bayes method, K nearest neighbor, support vector machine, etc. [5-8].

Naive Bayes email detection algorithm has extremely high detection accuracy and is widely used in text classification tasks, especially in spam detection due to its simplicity and efficiency [9, 10].

Corresponding author: 23104993d@connect.polyu.hk

The rest of this paper is organized as follows: Section 2 introduces the background knowledge of spam and the Naive Bayes classifier. Section 3 describes the dataset and the pre-processing of the data. Section 4 describes the work related to spam filtering using the Naive Bayes algorithm, including the implementation of the Simple Bayes classifier. Section 5 shows the results and analysis of the dataset the paper used. Finally, Section 6 summarizes this work and highlights future research directions

2 Method

2.1 Definition and characteristics of spam

Spam refers to a large number of emails sent without request, usually containing advertisements, malicious links, or other bad content. The characteristics of spam include:
-Mass sending: spam is usually sent in batches, with the aim of reaching as many recipients as possible.
-Diverse content: the content of spam can be advertisements, phishing links, malware, etc.

Highly concealed: Spammers usually use various technical means to circumvent email filtering systems [3, 4].

2.2 Basic principles of Naive Bayes classifier

In this essay, Naive Bayes is used to filter spam emails. Nevertheless, Naive Bayes made the assumption that each characteristic is independent of the others under conditional independence. The equation is:

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

Where the symbol P(A) stands for the probability of feature A, or the likelihood that feature A will manifest in the absence of any further information. To do more analysis, it must gather training samples of legitimate and spam emails. Condensed data sets from previous studies on the characteristics of legitimate and spam emails are available. The formula requires the initial spam probability, P(A), to be obtained.

The likelihood that attribute A (such the usage of a certain phrase) will be obvious when the email is spam is represented by the symbol P(A|B). It's the potential for event B to be used to deduce event A.

$$P(A|B) = \frac{P(AB)}{P(B)} \quad (2)$$

Where the event's marginal probability is denoted by P(B). A common term for P(B) is the prior probability. If no other information is available, it displays the likelihood that event B will take place. P(B|A), sometimes called "posterior probability," is a conditional probability that indicates the likelihood that event B will occur given the knowledge of event A. According to this theory, incident A occurs before event B [5].

2.3 Dataset source

The open-source project on GitHub, specifically located at https://github.com/Akashkg03/Spam-Email-Classification, provided the dataset utilized in this study. The primary purpose of this dataset, which was provided by user Akashkg03, is to practice and do research on spam categorization. Numerous email samples are included in the collection; each one has been meticulously labeled to indicate whether it is spam. These annotations are very helpful to re

searchers and developers as they provide a wealth of information for the training and assess ment of machine learning models, which helps in the creation and improvement of spam filt ering algorithms.

2.4 The work related to spam filtering using the Naive Bayes algorithm

To construct the model, this paper mostly utilizes the pre-made Jupiter notebook code from GitHub in this study. This research primarily examines how well naive Bayes performs in spam detection by looking at the program's output. But to support the experimental approach, this work will introduce the implementation of the naive Bayes classifier.

First, the likelihood of each word in spam and regular emails is determined using the naive Bayes algorithm. During this phase, the frequency of every word across many email categories is tallied, and its probability of being spam or ham is ascertained by using this frequency.

The chance of every word in the new email must then be multiplied to obtain an overall probability value, which is what needs to be calculated when categorizing a new email. Next, the likelihood of ham and spam is compared with this value.

Lastly, this paper identifies the new email category by contrasting the likelihood of spam and ham. It is categorized as spam if the likelihood of spam is higher; likewise, it is categorized as ham if the likelihood of spam is higher.

First, import the necessary libraries for both data processing and modeling. The next phase involves loading, cleaning, and preparing the datasets. Then, numerical features are constructed by extracting attributes from the text input. After that, the model will be trained using a classification strategy. Finally, evaluate the model’s effectiveness using the pertinent metrics. These methods enable the categorization of spam using a rudimentary Bayesian classifier [6].

3 Experimental result

3.1 Dataset description

The study's dataset comprises 5,575 emails, of which 4,827 are ham emails and 747 are spam emails. Email content and email classification, where spam emails have a category label of 1 and ham emails have a label of 0, are the two main features of the collection. In this experiment, 20% of the data is used as test data and 80% as training data.

According to statistics on email length, spam emails have an average length of 143.68 characters, which is much greater than the average length of ordinary emails, which is 75.47 characters. The average length of all emails is 84.60 characters 0 characters is the smallest email length, and 914 characters is the longest. This illustrates that spam emails usually include more material.

For the word frequency statistics, through the word frequency statistics of all emails, it can be found that the most common words include "ham", "i", "to", "you", "a", etc. In spam emails, the most common words include "spam", "call", "free", etc., reflecting the nature of promotion and advertising. In normal emails, the most common words are more daily and personal, such as "i", "you", "me", etc.

A further examination of the data set reveals significant differences between spam and regular emails in terms of word frequency, length, and other characteristics. These variations might serve as the foundation for screening and classifying spam. More studies on email sender traits and sending schedules might increase the precision of spam identification.

3.2 Dataset preprocessing

The following are the preprocessing steps for the experiment.

In the first step, it is necessary to preprocess each sample. The preprocessing steps include:
- Text cleaning: remove HTML tags, special characters, etc.
- Feature extraction: use the TF-IDF method to convert text into feature vectors.

The next step suggests using a laxer to split the sample into words and build a glossary. The easiest way to do this is to use the white space in the string summary as the word "boundaries".

Then it became possible to construct a one-hot vector for each word using Python techniques. This paper gets a word frequency vector by adding all the one-hot vectors together. From the point of view of cost-effectiveness, it is not included in this expository essay to consider the case of deactivated words.

According to the results obtained on the final dataset (see Table 1: Classification report below), the accuracy of the classifier using the Naive Bayes algorithm reached an extremely high 99.193%.

Table 1. Classification report

	Precision	**Recall**	**Numbering**	**Support**
0	0.99	1.00	1.00	970
1	0.97	0.97	0.97	145
accuracy			0.99	1115
macro average	0.98	0.98	0.98	1115
weighted average	0.98	0.99	0.99	1115

The examination and outcomes of the dataset show that the classification model does a good enough job of differentiating between spam and plain text.

How well the classification model separates spam from non-spam is demonstrated by the dataset and research findings.

For category 0 (non-spam), the model has an accuracy of 0.99, a recall of 1.00, an F1 score of 1.00, and a support of 970. This illustrates how almost flawless the model is at recognizing every non-spam message.

For category 1 (spam), the model has an F1 score of 0.97, an accuracy of 0.97, a recall of 0.97, and a support of 145. This provides more evidence of the model's robustness and spam-detection effectiveness.

4 Conclusions

The technique for categorizing spam emails distinguishes between legitimate and fraudulent emails with reasonable accuracy. However, because the model's evaluation depends on a specific dataset, its usage can be limited to other scenarios where spam categorization is crucial. This paper thinks this problem will be resolved in the future, enabling the application

of several state-of-the-art machine learning techniques to create spam classification systems that are more resilient and flexible.

Deep sequence modelling methods like as Recurrent Neural Networks and Long Short-Term Memory Networks, together with sophisticated feature extraction methods like Term Frequency-Inverse Document Frequency and Tokenizer, will be critical to future improvements. Through the identification of minute characteristics and intricate patterns present in email content, these strategies enhance the classification process's accuracy.

Methods for ensemble learning that make use of many model contributions can also greatly improve spam categorization. A more robust system may be created by combining the capabilities of many algorithms using strategies like stacking, boosting, and bagging.

Furthermore, without requiring labelled data, the inclusion of unsupervised learning techniques like autoencoders and clustering algorithms can aid in the detection of new and evolving spam trends. Keeping up with spammers might be quite helpful, as they frequently alter their tactics.

Finally, reinforcement learning allows the system to continuously learn from its performance and make changes, providing a dynamic method for spam identification. This might lead to the development of a proactive, flexible spam detection system.

The future of spam email categorization seems quite promising with the combination of many state-of-the-art approaches. A method that appears to hold promise is the use of transformer models, such GPT (Generative Pre-trained Transformer) and BERT (Bidirectional Encoder Representations from Transformers). In natural language processing applications, the models have proven to be exceptionally effective. These models increase spam identification accuracy because they more accurately represent the email's context and semantics.

Reference

1. S. Rushdi, M. Robet, Classification spam emails using text and readability features, IEEE 13th International Conference on Data Mining, IEEE, (2013), 5-7
2. M. A. Shafi'i, S. Maryam, O. Oluwafemi, et al, Comparative analysis of classification algorithms for email spam detection. I. J. Computer Network and Information Security. 10, 62-64 (2018)
3. N. Kumar, S. Sonowal, Email spam detection using machine learning algorithms, in Proceedings of the 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), IEEE, (2020), 108-113
4. Q. Yaseen, Spam email detection using deep learning techniques. Procedia Computer Science. 184, 853-858 (2021)
5. G. I. Webb, E. Keogh, R. Miikkulainen, Naïve Bayes. Encyclopedia of machine learning. 15, 713-714 (2010)
6. Y. Huang, L. Li, Naive Bayes classification algorithm based on small sample set, in Proceedings of the 2011 IEEE International Conference on Cloud Computing and Intelligence Systems, IEEE, (2011), 34-39
7. K. Agarwa, T. Kumar, Email spam detection using integrated approach of Naïve Bayes and particle swarm optimization, in 2018 Second International Conference on Intelligent Computing and Control Systems, IEEE, (2018), 685-690
8. G. Kang, K. Yusupov, M. R. Islam, K. Kim, K. Yim, The comparison of machine learning methods for email spam detection. in Proceedings of the Innovative Mobile and Internet Services in Ubiquitous Computing. 177, 45-56 (2023)

9. Koneru Anupriya, Kurakula Harini, Kethe Balaji, Karnati Geetha Sudha, Spam mail detection using optimization techniques. IIETA. 11, 78-89 (2021)
10. Vasudha Goswami, Vijay Malviya, Pratyush Sharma, Detecting spam emails/SMS using Naive Bayes, Support Vector Machine and other methods, ICCBI, Springer, (2020), 101-115