Spam filtering system based on machine learning

Wenhai Dong *

Department of Information Security, Beijing University of Technology, Beijing, China

* Corresponding Author Email: 13601217791@163.com

Abstract. With the process of the continuous growth of Internet technology, more and more people are participating in Internet activities. In daily email exchanges, there are often a certain amount of spam or uncivilized language that causes trouble to email users. However, since it is necessary to avoid misjudging normal emails as spam, it is extremely important to compare and analyze the accuracy results of different models. This paper employs various machine learning algorithms, such as support vector machines, random forest, naive Bayes to test spam detection, and discusses the optimal solution to this problem based on the results. Besides, A comprehensive explanation of the principles of various algorithms in the field of machine learning is provided in this paper. After performing necessary data processing and model handling, their advantages and disadvantages are analyzed to seek the optimal solution for the spam classification problem. Additionally, based on the existing results, this paper proposes future perspectives and possible improvement methods.

Keywords: Support Vector Machine; Random Forest; The Naive Bayes algorithm; Laplace smoothing.

1. Introduction

As the Internet expands in scale and significance, the volume and influence of online reviews are continually on the rise [1]. This could result in market manipulation, fake advertisements, counterfeit reviews, misinformation, and political astroturfing by spammers seeking to profit [2]. Simultaneously, when users are inundated with a high volume of email spam, the likelihood of overlooking important non-spam messages increases [3]. Many email users find themselves spending time deleting unwanted messages. Email spam can also incur costs for users with dial-up connections, consume bandwidth, and potentially expose minors to inappropriate content. In brief, in order to deal with these problems, Spam filtering system are built based on 3 different machine learning method.

This paper is structured in the following manner for the remaining sections: Section 2 describes 3 commonly used machine learning algorithms related to email spam filtering: SVM, Random Forest, and Naïve Bayes algorithm. Section 3 presents the specific processes and results of the email spam filtering experiments using data processing methods such as Laplace smoothing, along with corresponding result analysis. In the end, Section 4 wraps up this study and highlights potential avenues for future research.

2. Proposed approaches

2.1. Support Vector Machines

The binary classification model known as support vector machines (SVM) aims to identify a hyperplane that effectively separates different sample sets. Maximizing the margin is the fundamental principle behind the process of segmentation, which is ultimately converted into a convex quadratic programming problem to solve. In SVM, there is also the concept of soft interval. This is due to the appearance of outliers in the data set, which leads to poor generalization of the model. Therefore, a slack variable needs to be introduced to improve the generalization performance. Among them, $C>0$ is the penalty parameter, that is, the tolerance for error, which is a very important parameter of soft interval. The larger C is, the smaller the tolerance for outliers is. Only a few points can cross the interval boundary, and the model that tends to overfit will become complicated. Conversely, as the

value of C decreases, the model becomes more tolerant of outliers, allowing more data points to exceed the interval boundary, resulting in a smoother final model [4].

Furthermore, when SVM is combined with Term Frequency-Inverse Document Frequency (TF-IDF), the weighted vector machine gives class weights to the training samples to reflect the importance of different categories. For example, by increasing the weight of the common words in the email category, the number of misclassified samples in the category can be effectively reduced. The possibility of each email being correctly classified is increased, thereby improving the classification accuracy.

2.2. Random Forest

The random forest algorithm employs randomization to generate numerous decision trees [5]. In the forest, decision trees operate independently of one another, with each tree contributing to the overall output of the model. When a classification problem is addressed, each decision tree evaluates the test sample and assigns a category. The ultimate classification for the test sample is then established through a voting process that aggregates the outputs from all the decision trees. The randomness inherent in a random forest is evident in two key aspects:

I: Sample randomness, where a subset of samples is randomly chosen from the training set to serve as the root nodes for each decision tree;

II: Attribute randomness, which occurs during the construction of each decision tree, as a selection of potential attributes is randomly identified, and the most appropriate one is chosen to act as the split node.

2.3. The Naive Bayes algorithm

The Naive Bayes algorithm functions as a supervised learning method designed to tackle various classification challenges. It can be applied to determine outcomes such as customer churn, investment viability, credit ratings, and other multi-class classification issues. It is referred to as Naive Bayes because the probability calculations for each class are simplified to ensure they remain manageable [6]. This algorithm offers several benefits, including simplicity of comprehension, high efficiency in learning, and in specific areas, competitive performance compared to decision trees and neural networks for classification tasks. However, the accuracy of this algorithm is somewhat influenced by its underlying assumptions, which rely on the assumption that independent variables are conditionally independent and that continuous variables follow a normal distribution.

Naive Bayes classifiers are based on techniques derived from Bayesian classification methods [7]. The content of Bayesian decision theory is as follows: Suppose now there is a data set, which consists of two types of data, and the data distribution is shown in Fig. 1.

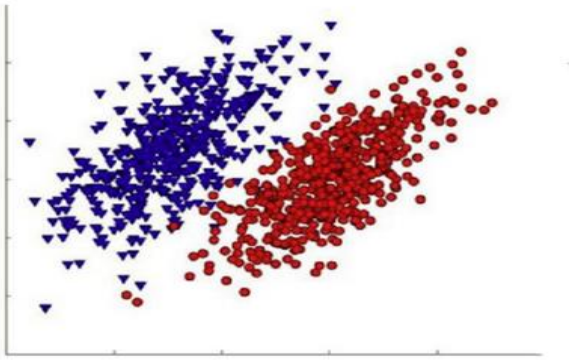

Fig 1. Example of data set.

In Fig. 1 (Example of Data Set), the probabilities p1(x,y) and p2(x,y) are defined to indicate the likelihood that a data point (x,y) belongs to category 1 (represented by the red dot) and category 2 (represented by the blue triangle), respectively. To classify a new data point (x,y), the following criteria can be applied:

If p1(x, y) exceeds p2(x, y), the data point is classified as belonging to category 1. Conversely, if p1(x, y) is less than p2(x, y), it is classified as belonging to category 2. In essence, the category with the highest associated probability should be selected. This principle encapsulates the fundamental concept of Bayesian decision theory, which revolves around opting for the decision that offers the greatest likelihood of accuracy. For the total probability formula is as follows:

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

P (A) is referred to as "prior probability," representing the assessment of the likelihood of event a occurring prior to the occurrence of event B. Meanwhile, P (A|B) is known as "posterior probability," indicating the updated evaluation of the likelihood of event an after event B has taken place. The term "likelihood function" refers to P (B|A)/P (B), which acts as a corrective factor that refines the estimated probability, bringing it closer to the actual probability.

This process exemplifies Bayesian inference. It begins with the estimation of a prior probability, followed by the incorporation of experimental results to assess whether these findings enhance or diminish the initial assessment. This evaluation leads to a posterior probability that is more aligned with reality. In this context, if the likelihood function P (B|A)/P (B) is greater than 1, it indicates an enhancement of the prior probability, thus increasing the likelihood of event A. Conversely, a likelihood function equal to 1 suggests that event B provides no additional insight into the probability of event A. Finally, if the likelihood function is less than 1, it implies a weakening of the prior probability, resulting in a decreased likelihood of event A.

3. Experimental results

3.1. Real-life datasets

A key factor in developing an effective model is the accuracy of the measurements [8]. To ensure that the model does not overfit, its performance can be assessed through k-fold cross-validation. This technique, using the same analysis method and the same tuning parameter values, allows for a comprehensive evaluation by partitioning the dataset into subsets, where the model is trained and tested multiple times, thereby providing a reliable measure of its effectiveness.

3.2. Model training

Specifically, in the case of spam detection, in order to block insulting speech, a quick filter is built. Messages that contain negative or insulting language will be labeled as inappropriate. The need to filter such content is a common requirement. To address this issue, two categories have been defined: insulting and non-insulting, with the former represented as 1 and the latter as 0.

Text is seen as word vectors or term vectors that is, converting sentences into vectors. Consider the words that appear in all documents, and decide which words should be contained in the vocabulary or the desired vocabulary set. For simplicity, it is assumed that the article has been segmented, stored in a list, and the vocabulary vectors are classified and labeled. Therefore, the naive Bayes classifier can be trained using the term vectors.

3.3. Laplace smoothing

However, when using the Bayesian classifier to classify documents, the product of multiple probabilities must be calculated to obtain the probability that the document is associated with a specific category. Naive Bayes cannot be used when the conditional probability is zero, in such cases, the predicted probability will also be zero [9]. In order to reduce this effect, the number of occurrences of all words can be initialized to 1 and the denominator can be initialized to 2. This practice is called Laplace Smoothing. In addition, another problem encountered is underflow, which is caused by multiplying too many small numbers. Anyone who has studied mathematics knows that when two decimals are multiplied, the number of times they are multiplied becomes smaller and smaller, which

causes underflow. In the program, rounding is performed at the corresponding decimal position, and the calculation result may become 0. To address this issue, the natural logarithm of the product result should be computed. Utilizing logarithms helps prevent errors that may arise from underflow or floating-point rounding. At the same time, there is no loss in processing using the natural logarithm.

3.4. Tuning parameters and test performance

From a content perspective, spam filtering can be classified as a "binary classification" problem, distinguishing between spam and legitimate emails. Therefore, various classification methods can be used for spam filtering. However, spam filtering is a classification problem in a specific field, which is different from general classification in at least the following aspects:

1. It is generally believed that users would rather receive more spam than misclassify legitimate mail as spam. Therefore, compared with the usual classification method, spam filtering pays more attention to the accuracy rate;

2. The environment in which spam filtering is implemented usually has higher performance requirements.

Therefore, the spam filtering method should focus on both the effect of implementation and the efficiency of implementation. For the parameter optimization of this experiment, a two-step method is used for data preprocessing. The first step is data cleaning: remove invalid data, duplicate data and missing values. The second step is text preprocessing: lowercase, remove punctuation, numbers and special characters. Currently, much of the real world’s data is incomplete and includes aggregated, noisy, and missing values [10]. So, some commonly used meaningless words (such as "is") should be removed. For each data, feature selection and extraction will be performed, term frequency-inverse document frequency can be used to characterize text data, which can help improve the model's effect on feature selection, or use statistical methods (such as chi-square test, information gain) to select important features and reduce the dimension of feature space. At the same time, select the smoothing parameter as 0.5 as described above to avoid the probability of zero caused by the absence of features in the training data.

3.5. Analysis of results

The experimental results are in Table 1. From the analysis, Naïve Bayes emerges as the most effective model overall, excelling in both accuracy and precision (0.9878 and 0.9889 respectively) while maintaining a competitive recall. Random Forest also shows strong performance with good accuracy and balanced precision and recall (0.9865 and 0.9846). In contrast, SVM, despite its high recall (0.9941), has lower overall effectiveness due to its reduced precision (0.9431) and accuracy (0.9671). Thus, for spam detection, Naïve Bayes is recommended for its balanced performance across key metrics.

Table 1. Experiment result.

Experiment Result				
	accuracy	precision	recall	F1 score
Random forest	0.9865	0.9846	0.9884	0.9865
SVM	0.9671	0.9431	0.9941	0.9679
Naïve bayes	0.9878	0.9889	0.9866	0.9878

4. Summary

The issue of email spam filtering is an important problem in information security and machine learning technology; experimental data indicates that the Naïve Bayes classifier is the best model for solving this type of problem. Nevertheless, the classification speed and accuracy of the Naïve Bayes classifier also depend on the dataset used. It performs better when the number of emails is small, allowing the Naïve Bayes classifier to demonstrate its efficient and quick performance, thereby effectively blocking more spam when the dataset comes from a single email account. Therefore, to

further enhance the effectiveness of the Naïve Bayes model in addressing this issue, future research can focus on adjusting parameters and improving the model. In-depth research into feature selection and extraction methods to better distinguish between spam and regular emails may be helpful in addressing such issues. For example, employing modern feature representation techniques such as TF-IDF or word embeddings can enhance the model's understanding of the text.

References

[1] Salloum S, Gaber T, Vadera S, et al. A systematic literature review on phishing email detection using natural language processing techniques. IEEE Access, 2022, 10: 65703-65727.

[2] Rao S, Verma A K, Bhatia T. A review on social spam detection: Challenges, open issues, and future directions. Expert Systems with Applications, 2021, 186: 115742.

[3] Ahmed N, Amin R, Aldabbas H, et al. Machine learning techniques for spam detection in email and IoT platforms: analysis and research challenges [J]. Security and Communication Networks, 2022, 2022(1): 1862888.

[4] Pisner D A, Schnyer D M. Support vector machine [M]//Machine learning. Academic Press, 2020: 101-121.

[5] Karabadji N E I, Korba A A, Assi A, et al. Accuracy and diversity-aware multi-objective approach for random forest construction. Expert Systems with Applications, 2023, 225: 120138.

[6] Viet T N, Le Minh H, Hieu L C, et al. The Naïve Bayes algorithm for learning data analytics. Indian Journal of Computer Science and Engineering, 2021, 12(4): 1038-1043.

[7] Chen H, Hu S, Hua R, et al. Improved naive Bayes classification algorithm for traffic risk management. EURASIP Journal on Advances in Signal Processing, 2021, 2021(1): 30.

[8] Ligthart A, Catal C, Tekinerdogan B. Analyzing the effectiveness of semi-supervised learning approaches for opinion spam classification. Applied Soft Computing, 2021, 101: 107023.

[9] Burqan A, El-Ajou A, Saadeh R, et al. A new efficient technique using Laplace transforms and smooth expansions to construct a series solution to the time-fractional Navier-Stokes equations. Alexandria Engineering Journal, 2022, 61(2): 1069-1077.

[10] Bischl B, Binder M, and Lang M, et al. Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges [J]. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2023, 13(2): e1484.