Report Yiyi Wang

1. Background

In today's digital age, spam filtering has become an important task to protect users' mailboxes and improve information security. With the rapid development of the Internet, the number and types of spam are increasing, which brings a lot of troubles to users and enterprises. Traditional spam filtering methods, such as naive Bayes algorithm, have achieved good results in the past research because of its simplicity and high efficiency. However, with advances in natural language processing techniques, algorithms based on large language models (LLMS) have come to the fore, demonstrating greater semantic understanding and contextual processing capabilities. These modern algorithms, through deep learning techniques, are able to capture more complex text features, thus improving the accuracy and robustness of spam classification. Therefore, discussing the advantages and disadvantages of naive Bayes algorithm and large language model in spam filtering can not only provide a new perspective for the research of related fields, but also provide a reference for the selection of practical applications.

In our daily life, we often receive all kinds of spam, such as advertisements from merchants, discount promotion information, mail, financial promotion information, etc. Generally speaking, the mail client will set certain keywords to shield this spam, or classify the mail, but there will always be some fish that escape the net. Multinomial naive Bayes, Bernoulli naive Bayes, supplementary naive Bayes, logistic regression, support vector machine, KNN, decision tree, random forest, gradient lift, neural network (multi-layer perceptron) algorithms are used for spam classification, test and compare the performance of different algorithms, and select the algorithm suitable for spam classification.

When classifying English user comments, the accuracy and precision of the Multinomial Naive Bayes classifier are higher than those of the Bernoulli Naive Bayes classifier.[1] The Multinomial Naive Bayes classifier has higher values for the recall parameter of positive comments compared to the Bernoulli Naive Bayes classifier.[2]

2. Data Resource

2.1 Trec06c

Trec06c is a public spam corpus, provided by the International Text Retrieval Conference, which is divided into English data sets (trec06p) and Chinese data sets (trec06c). The emails contained in TREC06C are from real emails and retain the original format and content of emails. Here I select the English datasets.

Total number of Email: 37313
Spam Email: 24542
Normal Email: 12771
size of training set: 27984
size of test set: 9329

2.2 SMS Spam Collection

The SMS Spam Collection is a public set of SMS labeled messages that have been collected for mobile phone spam research.

Total number of SMS: 5572
Spam SMS: 747
Normal SMS: 4825
size of training set: 4179
size of test set: 1393

3. Method

TREC's data set is provided by file, with spam and ham marked by an index for each message in a file. The following code extracts the body of the message and outputs all the body and tags of the message to a file for the next step in converting the bag model. The train_test_split() function provided by sklearn splits the data into training and test sets. Bag of words model can translate a sentence into a vector representation, is a relatively simple and straightforward method, it does not consider the order of words in the sentence, only consider the number of words in the vocabulary in the sentence. stop_words= 'english' means to use the built-in English stop words, such as a, the article is very high frequency but ineffective for analysis, so it is simply removed.

First, fit_transform(x_train) is used to construct the word frequency matrix according to the stop word list, and then transform(x_test) is used to calculate the word frequency matrix of the test set according to the word order of the constructed word frequency matrix.

Finally, with the training and testing sets from TREC and SMS Spam, we test and compare the Multinomial naive Bayes, Bernoulli naive Bayes, complementary naive Bayes, logistic regression, support vector machine, KNN, decision tree, random forest, gradient lift, neural network (multi-layer perceptron) algorithms for spam classification, the performance of different algorithms. With the performance evaluation results, we can select the algorithm suitable for spam classification according to different size of datasets.

For evaluation, I selected five parameters.

Precision (P): Spam detection rate. The higher the accuracy rate, the less spam "escapes". Intuitively, precision P is the classifier's ability not to label negative samples as positive, while recall R is the classifier's ability to find all positive samples.

Recall rate (R): Spam detection rate. This indicator reflects the ability of the filtering system to detect spam, and the recall rate reflects the ability of the filtering system to "find the right spam". The larger the recall rate, the less likely it is to misjudge legitimate emails as spam.

Accuracy: The accuracy rate of all mail (both spam and legitimate mail). Accuracy=(TP+TN)/N

F1 metric: F1 is actually a harmonic average of recall rate and correct rate. It combines recall rate and correct rate into one metric. The larger the F1, the better the model.

TCR: we don't want to mistake legitimate mail for spam. In order to represent the cost of spam system in different cases, we can use the concept of cost factor. Assume that the loss of misidentifying legitimate mail as spam is λ times that of misidentifying spam as

legitimate. For example, $\lambda = 9$ means that a legitimate email error is nine times more costly than a spam error.

4. Results

Trec06c	F1 Score	Precision	Recall	Accuracy	TCR Score
Multinomial Naive Bayes	0.98	0.99	0.96	0.97	12.2
Bernoulli Bayes	0.93	0.88	0.99	0.9	7.2
Complement Bayes	0.98	0.99	0.95	0.97	11.2
KNN Algorithm	0.96	0.92	0.99	0.94	11.2
SVN Algorithm	0.99	0.99	0.99	0.99	48.74
Decision Tree	0.98	0.97	0.99	0.97	24.28
Random Forest	0.99	0.99	0.99	0.99	45.03
Gradient Boosting	0.96	0.93	0.99	0.94	12.49
Netural Network	0.99	0.99	0.99	0.99	75.09

SMS Spam	F1 Score	Precision	Recall	Accuracy	TCR Score
Multinomial Naive Bayes	0.95	0.96	0.93	0.98	6.03
Bernoulli Bayes	0.88	1	0.78	0.97	1.83
Complement Bayes	0.89	0.84	0.95	0.97	4.39
KNN Algorithm	0.78	1	0.65	0.95	0.93
SVN Algorithm	0.92	0.99	0.87	0.98	3.34
Decision Tree	0.88	0.91	0.85	0.96	2.45
Random Forest	0.9	1	0.82	0.97	2.42
Gradient Boosting	0.83	0.99	0.72	0.96	1.29
Netural Network	0.93	0.97	0.9	0.98	4.32

5. Analysis and discussion

From the results from part 4, we can find in naïve Bayes methods that the best is multinomial model naive Bayes classifier, followed by complementary naive Bayes classifier, and the worst is Bernoulli naive Bayes classifier. When TREC data sets with more samples are used, the classification effects of Bernoulli naive Bayes and complementary naive Bayes are close, and the increase of data sets significantly improves the classification effect of Bernoulli naive Bayes.

As we know, complementary naive Bayes is an adaptive algorithm of the standard multinomial naive Bayes algorithm, which is especially suitable for unbalanced data sets. Specifically, complementary naive Bayes uses supplementary statistics from each class to calculate the weight of the model. The inventors of complementary naive Bayes have empirically shown that the parameter estimation of complementary naive Bayes is more stable than that of multinomial naive Bayes. In addition, complementary naive Bayes often outperforms complementary naive Bayes in text classification tasks.

Thus, we can make the comparison of multinomial model and Bernoulli model. The Bernoulli model does not consider the number of occurrences of terms, while the multinomial model does. The Bernoulli model is suitable for processing short documents, while the multinomial model is suitable for processing long documents. The Bernoulli model performs better when the number of features is small, while the multinomial model performs better when the number of features is large.

For multinomial model, the influence of word frequency is fully considered. Obviously, the higher the word frequency of a word in an article, the more representative it is, so the influence of word frequency should be considered. However, this does not avoid the impact of extreme data, such as 100 studies in a row, and many other

rows do not have studies. For Bernoulli model, it is easier to implement, can reduce the impact of the above extreme data

Also, in the spam filtering task, a large number of experiments show that linear kernel has high performance, which is similar to that of other kernel functions. Suitable feature representation of linear kernel can obtain better classification performance and lower computational complexity. However, if the feature dimension is much larger than the number of samples, SVM is mediocre. When the sample size is very large and the kernel mapping dimension is very high, SVM is not suitable for use.

At the same time, KNN is the most commonly used instance-based method. KNN has no training process, and compares the text to be classified directly with each text in the training set, and then obtains the category of the new text according to the most similar k text. The principle of KNN is very intuitive. KNN often achieves better results in text classification. However, due to the limitation of its classification speed, it is not suitable for spam filtering which requires high classification speed. Nevertheless, for research purposes, some literature still applies it to the field of spam filtering. KNN has similar performance to naive Bayes when the sample set is larger, but KNN classification speed is slower than naive Bayes.

Finally, it can be seen that neural network has better classification effect when the data set is larger, and multinomial naive Bayes has faster classification speed and better classification results when the data set is smaller. Using neural network for classification can also update the model incrementally, which is suitable for training certain samples and then obtaining more accurate classification results through users manually marking spam. But for the shortcoming, MLP non-convexity loss function with hidden layer where multiple local minima exist. Therefore, different random weight initializations may result in different precision. MLP requires adjustment of many hyperparameters, such as the number of hidden neurons, the number of layers, and the number of iterations. MLP is sensitive to feature scaling.

Reference

[1] N. L. Octaviani, E. Hari Rachmawanto, C. A. Sari and I. M. S. De Rosal, "Comparison of Multinomial Naïve Bayes Classifier, Support Vector Machine, and Recurrent Neural Network to Classify Email Spams," 2020

[2] S. Rastogi, R. Sambyal, P. Tyagi and R. Kushwaha, "Multinomial Naive Bayes Classification AlgorithmBased Robust Spam Detection System," 2024

[3] N. Kalcheva, G. Marinova and M. Todorova, "Comparative Analysis of the Bernoulli and Multinomial Naive Bayes Classifiers for Text Classification in Machine Learning," 2023