E-mail Spam Classification using KNN and Naive Bayes

Qianhe Ouyang[1, *], Jiahe Tian[2] and Jiale Wei[3]

[1]University of Electronic Science and Technology of China, Chengdu, China

[2]Nanjing Forestry University, Nanjing, China

[3]Xi'an University of Technology, Xi'an, China

*Corresponding author: 2020090904016@std.uestc.edu.cn

Abstract. E-mail spam filtering is becoming a critical and concerned issue in network security recently, and multiple machine learning techniques have been applied to tackle such sort of classification problem. With the emerging of machine learning framework, most of the tasks has been changed via the effective machine learning algorithms with satisfying performance and high speed. However, the underlying performances of different algorithms under certain given circumstances still lack of an intuitive demonstration. Hence, this study mainly focuses on the performance of two widely-used algorithms (KNN and Naive Bayes) from metrics including accuracy and running time, comparing the unique advantage of each algorithm when classifying emails. The paper uses thousands of spam data to feed two algorithms and analyzes both results respectively, indicating that KNN classifier performs better when determining the spam messages while the opposite is true for the Naive Bayes classifier. Thus, designers can pick an appropriate algorithm easily when dealing with spam filter issues under a given dataset whose features and properties are known.

Keywords: Spam filter; K-Nearest Neighbor; Naive Bayes.

1. Introduction

Nowadays the volume of emails is growing rapidly as emails represent a primary, fast, and cheap communication tool in all fields. The increased use of e-mail also entails more spam attacks for Internet users. Spam can be sent from anywhere on the planet from users having deceptive intentions that have access to the Internet. Spams are unsolicited and unwanted emails sent to recipients who do not want or need them. These spam emails have fake content with mostly links for phishing attacks and other threats, and these emails are sent in bulk to a large number of recipients [1]. Accordingly, the need for more accurate spam filters has been raised. It is imperative to detect spam emails in near real time to have an effective and secure email filter. It becomes a security challenge to detect that threat [2].

Amiza Amir's research team used a variety of spam processing algorithms to evaluate their systems. They have evaluated their system against centralized state-of-the-art algorithms (NN, KNN, Naive Bayes, BPNN, and RBFN) and distributed P2P-based algorithms (Ivote-DPV, ensemble KNN, ensemble naive Bayes, andP2P-GN) [3]. The experimental results show that their method is highly accurate with a 98 to 99% accuracy rate, and incurs a small number of messages—in the best-case, it requires only two messages per recall test.

According to the research idea of Amir, we decided to focus on discussing and comparing the effectiveness and success rate of KNN and naive Bayes in spam filtering. First of all, the Naive Bayesian model originates from classical mathematical theory and has stable classification efficiency and good performance for small-scale data, can handle multi-classification tasks, suitable for incremental training, especially when the amount of data exceeds memory, we can batch to incremental training. It is not sensitive to missing data, and the algorithm is relatively simple, so it is often used in text classification. At the same time, KNN algorithm has high accuracy, is insensitive to abnormal values, and has no data input assumption. However, this algorithm has high time complexity and space complexity. Because of the high popularity and wide application of these two algorithms in machine learning, our group decided to apply the two algorithms (including Naive Bayes and KNN) to spam filtering and compared the performance of the two algorithms in this aspect.

we get the results that each of the two algorithms has its own advantages and disadvantages in mail filtering. In the experiment, we find the Naive Bayes is better than KNN according to the mean accuracy with more fast inference time. Furthermore, KNN has the potential in detecting spam e-mails.

The rest of the paper is organized as follows: Section 2 mainly describes the data processing methods that can be used and the problems encountered in the process of data processing, and expounds and analyzes the principles of naive Bayes and KNN. In Section 3, two different algorithms (Naive Bayes and KNN) are used to calculate and process a certain database, and the processing data (accuracy, running time, etc.) of the two algorithms in spam filtering are obtained. Finally, the two algorithms are compared and discussed. Finally, Section 4 concludes the work and highlights the direction for future research.

2. Method

2.1. Data processing

For the dataset [4], part of the Enron-Spam datasets is chosen to be implemented in the Naive Bayes and KNN algorithms. There exist six groups of Enron-spam datasets with six different ham-spam proportions respectively. The first section of the dataset is selected in the project, containing 3672 ham emails as well as 1500 spam emails, and the ham-spam proportion is 3:1 approximately. Each ham has a different date ranging from 1999-12-10 to 2002-01-11 and 2003-12-18 to 2005-09-06 for the counterpart.

The major challenge when analyzing a certain mail content is feature selection. And for simple mail context, each word matches a unique feature correspondingly. Thus the feature is constantly represented in the form of objects like vectors of numbers or feature vectors [5]. There are two main sorts of selection methods currently. One is to simply take every word in the mail into consideration; the other is to select part of the words and extract its features. The first approach indicates that there's no necessity to select certain features based on certain metrics, but the time and space expense is not expected to be neglected, affecting the efficiency when implementing algorithms to some extent. Another approach is to select the most weighing features among words, which means that the word tends to appear with the highest frequency or it might have the most biased possibility of constituting spam or ham [6]. However, there may exist potential risks that loss of information may occur during the conversion process, and spam and ham emails with the identical feature vector will be incorrectly classified, resulting in a high false positive rate [7].

Overall, the time and space that the first approach takes are acceptable for this project. Therefore, all the appearing words are considered as features for both Naive Bayes and KNN algorithms.

2.1.1. Data processing for Naive Bayes

Specifically, for the Naive Bayes algorithm, two dictionaries are created in order to store the words scanned from the training data: ham_dict and spam_dict. After that, the testing process starts with initialization that assigns equal possibility (0.5) to both ham and spam. For the next few ongoing mails, they are split into words and predicted based on the occurrence recorded in the two dictionaries above. As a result, the likelihood which is recorded in variables num_ham and num_spam can be computed as follow: (the probability of ham can also be calculated similarly)

$$\mathrm{p(spam)} = \frac{\mathrm{num\_spam}}{\mathrm{num\_ham+num\_spam}} \tag{1}$$

Finally, the possibility of the whole email being spam or ham can be computed by multiple prior. If the possibility of a certain mail being spam is greater, it's believed to be spam and recorded.

2.1.2. Data processing for KNN

The data processing for KNN algorithm is similar to that of Naive Bayes, but some enhancements are applied to the algorithm when computing the K Nearest Neighbors.

As mentioned above, the function of dictionaries is to record every word that appears in the mail content. If the length of the dictionary is N, an N-length vector is created in which every index of the vector matches a single word. The index value will be set to 1 if the word appears in the email and 0 otherwise.

When it comes to the distance calculation in KNN algorithm, the following formula is adopted:

$$\text{Cosine Distance}(i,j) = \frac{v_i\, v_j}{|v_i|\,|v_j|} \tag{2}$$

For the dictionary, the length N could be more than thousands, the time that computing the distance between an unknown email to all other training emails takes is unacceptable. Thus, by computing the similarity indirectly can the main idea of KNN methods be accomplished, instead of adopting distance conventionall. If two emails have greater similarity, it's believed that their distance is smaller. The formula below defines similarity [8]:

$$\text{Cosine Similarity}(i,j) = \frac{C}{\sqrt{A*B}} \tag{3}$$

A: the number of terms in email i

B: the number of terms in email j

C: the number of terms that email i and j both have

The overall procedures of data processing can be conducted following the steps below [9]: (1) For every email in the testing set, split it and get the words in it. split every email in the testing set to obtain the words (2) Compute the similarity of this email to any other emails in the training set (3) Sort the training set using similarity and select the K nearest neighbors (4) Classify the email as the category that has the most number of neighbors.

2.2. Naive Bayes

The proposed approach mainly consists of three basic phases from the data input to the final classification result output. These phases are explained in the following [10].

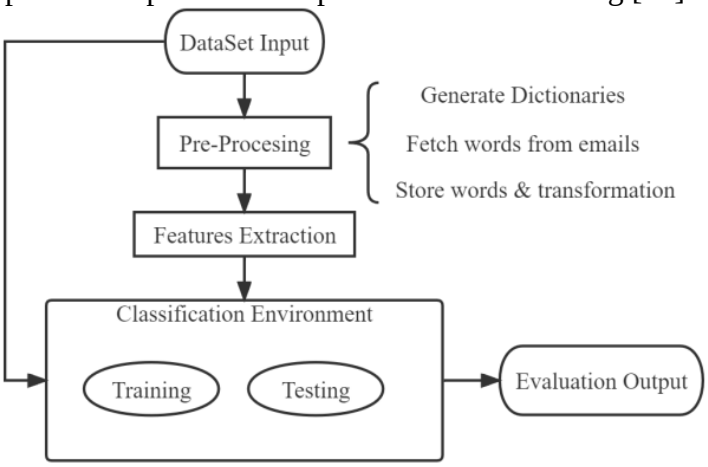


Fig.1 Spam-Filter procedures

The Bayesian statistical theory is to predict the probability of a test roll occurring based on the time it has already occurred. In the assumptions of Bayesian theory, if the outcome of an event is uncertain, the way to quantify the event is its occurrence probability. If the probability of occurrence of events in past tests is known, then the probability of occurrence of events in future tests can be calculated according to mathematical probability theory.

For the Naive Bayes classifier, the motivation is to learn the class's conditional density which is p(x|y) for all the values of y and also learn the class prior p(y). This classifier works on the Bayes theorem stated as [11]:

$$p(y|x) = \frac{p(x,y)}{p(x)} = \frac{p(x|y)p(y)}{\sum_{k\in C} p(x|k)p(k)} \tag{4}$$

p(x): data probability.

p(y): probability of a hypothesis for being true, which is also referred to as prior probability of y.

p(y|x):probability of hypothesis y given the data x, which is also referred to as posterior probability.

p(x|y):probability of data x given the hypothesis that y is true.

Naive Bayes classifier also assigns the labels by maximizing the probability p(y|x) formally defined as [11,12]:

$$f(x) = \arg\max_y p(y|x) = \arg\max_y \frac{p(x|y)p(x)}{p(x)} \tag{5}$$

Due to the reason that p(x) is not depended on the class, the function in Equation (5) can be rewritten as:

$$f(x) = \arg\max_y p(x|y)p(y) \tag{6}$$

Applying the theories above to the spam filter project, p(x) is the probability of mails $x = \{x_1,x_2,x_3,...,x_m\}$ where the x_m is the frequency of word m. If unbiased estimators are given, the probability p(y) can be written as:

$$p(y = k) = a_k, \text{ for } \forall k \in C \tag{7}$$

a_k: the frequency of the class k

C: the set of labels{0,1}

What can't be neglected is that the assumption is considered that all x_i are independent to the given y. Thus, the constitution of the words in a given class is independent and identically distributed. In order to enable the $p(x_i|y)$ to be various enough, there's a necessity to assign an appropriate distribution. The Gaussian distribution is adopted to $p(x_i|y)$:

$$x|y = 0 \sim N(\mu_0, \Sigma_0)$$

$$p(x|y = 0) = (2\pi|\Sigma_0|^2)^{-\frac{1}{2}}\exp\{-\frac{1}{2}(x-\mu_0)^T|\Sigma_0|^{-1}(x-\mu_0)\}$$

$$x|y = 1 \sim N(\mu_1, \Sigma_1)$$

$$p(x|y = 1) = (2\pi|\Sigma_1|^2)^{-\frac{1}{2}}\exp\{-\frac{1}{2}(x-\mu_1)^T|\Sigma_1|^{-1}(x-\mu_1)\} \tag{8}$$

For the max probability estimator for μ_0,μ_1 and Σ :

$$p(y = k) = \frac{1}{n}\sum_{i=1}^{n} 1\{y^{(i)} = k\} = a_k$$

$$\mu_0 = \frac{\sum_{i=0}^{n} 1\{y^{(i)} = 1x^{(i)}\}}{\sum_{i=0}^{n} 1\{y^{(i)} = 1\}}$$

$$\Sigma_0 = \frac{1}{n}\sum_{i=1}^{n}(x^{(i)}-\mu_0)(x^{(i)}-\mu_0)^T$$

$$\mu_1 = \frac{\sum_{i=1}^{n} 1\{y^{(i)} = 1x^{(i)}\}}{\sum_{i=1}^{n} 1\{y^{(i)} = 1\}}$$

$$\Sigma_1 = \frac{1}{n}\sum_{i=1}^{n}(x^{(i)}-\mu_1)(x^{(i)}-\mu_1)^T \tag{9}$$

The basic property of the naive Bayes classifier is to assign labels to maximize the probability p(y=k|x) by taking cheap computation such as p(y = 0), p(y = 1), μ_0, μ_1, Σ_0, and Σ_1 [11,12].

The assumption that p(x|y) is a multivariate Gaussian can be verified through normality tests implemented in R, i.e., Mardia's and Henze-Zirkler's tests [13].

Finally, in this case, the Naive Bayes algorithm is implemented by using the Gaussian distribution. The code implemented is based on the following equation [14]:

$$p(x|y = k) = \prod_{i=1}^{p}[(2\pi\sigma_{ki}^2)^{-\frac{1}{2}}\exp\{-\frac{1}{2\sigma_{ki}^2(x_i-\mu_{ki})^2}\}] \quad (10)$$

2.3. K-Nearest Neighbor

The spam filter using KNN algorithm is also implemented in the procedures shown in Fig.1. K-Nearest Neighbour is a widely used classifier in machine learning mostly in the field of image and video recognition. In this project, however, it's adopted to tackle classification issues.

When it comes to the features of KNN, non-parametric and slow learning techniques are two main factors that should be taken into consideration. It is non-parametric in a sense that it does not work on the assumptions related to the mathematical theory. The model determined from the data does not have to train itself from the data points which makes the training faster but slows down the testing phase. Therefore, KNN requires huge memory, computational cost, and low tolerance to noise [15].

The value of K is a critical factor in the algorithm because the classification depends upon the (odd)number of neighbors. And The classification will be determined by the majority of votes from its K neighbors. To calculate the closest points, several types of distances are defined as follows:

$$\text{Minkowski Distance:}(\sum_{i=1}^{n}|x_i - y_i|^p)^{1/p} \quad (11)$$

Specifically, equation (11) is known as Mahattan Distance when p=1 ($\sum_{i=1}^{n}|x_i - y_i|$) and Euclidean Distance when p=2 ($\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$).

The algorithm simply follows some standard basic steps: (1) calculate the distance (2) find the closest neighbors (3) vote for the labels. Afterward, whenever a new data point enters the dataset, the algorithm has to assign its class(A, B).

3. Experiment

3.1. Comparison results

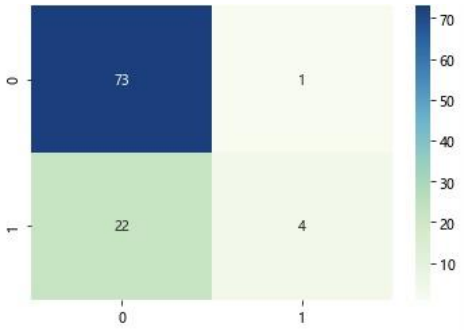
In the test, there are 2577 data chosen from the database for Naive Bayes and 100 data chosen from the database for KNN. The experimental data covers 100 test results of Naive Bayes as well as KNN. The data result of Naive Bayes shows in Appendix and KNN shows in Appendix B. Each data result has five columns. The first column of data shows the total number of emails that the program produced. The second column of data shows the total number of emails that the spam filter succeed to resolve in each round of testing, and the third column of data shows the number that the filter failed to resolve. The fourth column of data shows each probability that the spam filter succeeds to distinguish a message in every round of test data and the fifth column shows the running time for the whole filter to produce all of the tested data.

Table 1. Naive Bayes and KNN comparing

Type	Mean		Variance	
Name	Accuracy	Time	Accuracy	Running time
Naive Bayes	91.32%	0.359ms	0.00465	0.00123
KNN	76.98%	120.34ms	0.00175	0.37249

Table 1 shows the average and the variance of accuracy and running time of Naive Bayes and KNN. It is obviously that the average accuracy of Naive Bayes is higher than KNN and the average running time of Naive Bayes is lower than KNN. The variance of accuracy of Naive Bayes is better than KNN while the variance of running time of KNN is lower than Naive Bayes.It shows that the accuracy of Naive Bayes is higher and more stable. The running time of KNN is more stable but higher than Naive Bayes.

3.1.2 Discussion



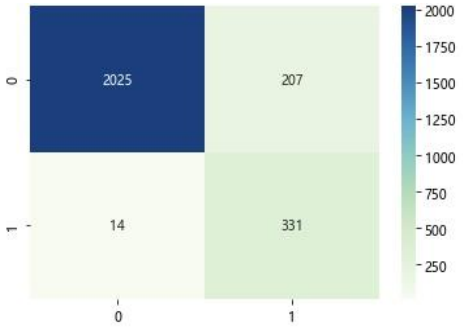


Fig.2 Confusion matrix of KNN **Fig.3** Confusion matrix of Naive Bayes

Related to these two matrices, here are some discussions about Naive Bayes and KNN. As for forecasting the e-mails which are spam, the accuracy of Naive Bayes is 61.52% and the accuracy of KNN is 80%. As for the e-mails which are ham, the accuracy of Naive Bayes is 99.31%, the accuracy of KNN is 76.84%. According to the confusion matrix and the accuracy of each model, it is obvious that the ability for Naive Bayes to forecast ham e-mails is better than KNN, and the ability for KNN to forecast spam is better than Naive Bayes. Thus, it is better to use the Naive Bayes model to distinguish the ham e-mails and use the KNN model to distinguish the spams.

4. Conclusion

The performance of Naive Bayes and KNN classifiers are based on use cases and different datasets. This study clearly reveals that both algorithms are suitable to be chosen when filtering spam emails. Datasets that have fewer instances of e-mails and attributes can perform precisely for both classifier; KNN has the highest percentage when filtering the spam messages while Naive Bayes has the highest percentage when determining the ham messages; Naive Bayes has a much lower running time flitting messages. The major contribution and motivation of this study is to provide appropriate suggestions with those who desire to pick a certain algorithm based on performance concerns. By comparing the accuracy as well as running time, it would be easier to make estimations towards different circumstances. Currently, this study introduces two kinds of algorithms without others like Support Vector Machine being taken into consideration. Also, the variants of KNN(KD-tree, ball-tree, LSH, etc.) may perform improved different predictions, which also require future research which might be more complicated.

References

[1] Siddique, Z. B., Khan, M. A., Din, I. U., Almogren, A., Mohiuddin, I., & Nazir, S. (2021). Machine learning-based detection of spam emails. Scientific Programming, 2021.

[2] Magdy, S., Abouelseoud, Y., & Mikhail, M. (2022). Efficient spam and phishing emails filtering based on deep learning. Computer Networks, 206, 108826.

[3] Amir, A., Srinivasan, B., & Khan, A. I. (2018). Distributed classification for image spam detection. Multimedia Tools and Applications, 77(11), 13249-13278.

[4] Wander Fernandes Junior. Enron-Spam dataset. 2019. Retrieved on August 8, 2022. Retrieved from: https://www.kaggle.com/datasets/wanderfj/enron-spam

[5] Peng, W., Huang, L., Jia, J., & Ingram, E. (2018, August). Enhancing the naive bayes spam filter through intelligent text modification detection. In 2018 17th IEEE international conference on trust, security and privacy in computing and communications/12th IEEE international conference on big data science and engineering (TrustCom/BigDataSE) (pp. 849-854). IEEE.

[6] Tretyakov, K. (2004, May). Machine learning techniques in spam filtering. In Data Mining Problem-oriented Seminar, MTAT (Vol. 3, No. 177, pp. 60-79). Citeseer.

[7] Aas, K., & Eikvil, L. (1999). Text categorisation: A survey. Technical report, Norwegian computing center.

[8] Soucy, P., & Mineau, G. W. (2001, November). A simple KNN algorithm for text categorization. In Proceedings 2001 IEEE international conference on data mining (pp. 647-648). IEEE.

[9] Firte, L., Lemnaru, C., & Potolea, R. (2010, August). Spam detection filter using KNN algorithm and resampling. In Proceedings of the 2010 IEEE 6th international conference on intelligent computer communication and processing (pp. 27-33). IEEE.

[10] Deshmukh, N., Dhumal, V., Gavasane, N., & Jadhav, S. V. (2021). Spam Detection by Using Knn Algorithm Techniques. Int. J, 6, 27-33.

[11] Almeida, T. A., & Yamakami, A. (2012). Advances in spam filtering techniques. In Computational Intelligence for Privacy and Security (pp. 199-214). Springer, Berlin, Heidelberg.

[12] Hovold, J. (2005, July). Naive Bayes Spam Filtering Using Word-Position-Based Attributes. In CEAS (pp. 41-48).

[13] Korkmaz, S., Göksülük, D., & Zararsiz, G. Ö. K. M. E. N. (2014). MVN: An R package for assessing multivariate normality. R JOURNAL, 6(2).

[14] Williams, C. K., & Barber, D. (1998). Bayesian classification with Gaussian processes. IEEE Transactions on pattern analysis and machine intelligence, 20(12), 1342-1351.

[15] Barigou, F., Beldjilali, B., & Atmani, B. (2014). Using cellular automata for improving knn based spam filtering. Int. Arab J. Inf. Technol., 11(4), 345-353.