Recognition of Spam Messages Based on Text Mining

Yajie Fang and Ping Zhang*
Department of information management and information system, Shanghai Business School, Shanghai, China
*Corresponding author email: 21140007@sbs.edu.cn

Abstract. With the rapid development of science and technology, the application of artificial intelligence technology is becoming more and more extensive. The maturity of artificial intelligence technology is inseparable from big data and machine learning. Big data is the foundation of artificial intelligence, while machine learning is the core. To mine the value hidden in big data, in order to achieve text classification, data prediction, and provide strategic decision basis, we need the power of Machine Learning. This article will introduce the classic algorithm in machine learning, Naive Bayes, KNN nearest neighbor, and support vector machine algorithm, and apply them to text mining of spam text sets. It shows that Machine Learning is the fundamental way to make machines as intelligent as humans.

Keywords: Machine learning; Text mining; Naive Bayes; KNN; SVM.

1. Introduction
Machine learning is a cross-disciplinary subject, which is closely related to many subjects such as probability theory, statistics, and computer science. The key of machine learning is learning, and the purpose is to enable computers to realize the behavior of human learning in order to acquire new knowledge and skills to improve their performance.

![AI & Machine Learning](image)

Figure 1. AI & Machine Learning

Machine learning algorithms are mainly divided into four categories, classic machine learning, reinforcement learning, neural networks and deep learning, integrated methods.

2. Algorithms of Machine Learning
This article introduce supervised machine learning, which is based on known features for classification and regression prediction.

2.1. Naive Bayes Algorithm
Naive Bayes algorithm is one of the most classic algorithms in machine learning, and it plays an important role in the fields of text classification, character recognition and image recognition.
The algorithm is based on the assumption that "feature attributes are independent of each other." Its core idea is to find the probability of an item to be classified in each category. Among the obtained probabilities, the category corresponding to the maximum value is the category to which the item to be classified belongs. The algorithm flow is as follows.

**Figure 2.** The Naive Bayes algorithm flow

First, the characteristic attribute X must be determined. This step is the only step in the entire Naive Bayes algorithm that needs to be done in manual. It is necessary to manually determine the attributes of the features according to the specific situation, and appropriately divide each feature, and then to classify this part of the classification.

There is a sample data set D to be classified, and its classification feature attribute set is $X = \{x_1, x_2, ..., x_n\}$, the prior probability of the training sample set $y_i$ can be estimated by the frequency of occurrence of various samples, or calculated by maximum likelihood estimation. However, the maximum likelihood estimation has a flaw. When the number of occurrences of an event is zero, it will have a probability value of zero. Therefore, we usually use Bayesian estimation to calculate the prior probability.

\[
P(Y = C_k) = \frac{\sum_{i=1}^{m} I(y_i = C_k) + \lambda}{m + K\lambda}, \quad k = 1, 2, ..., K, m \text{ represents each sample of } m, \text{ and } K \text{ is the number of categories.}
\]  

Next step is to calculate the conditional probability $P(X=x_j|Y=C_k)$ of the characteristic attribute $x_j$ in each category, that is, the frequency of the characteristic attribute $x_j$ in the sample category $C_k$.

\[
P(X = x_j | Y = C_k) = \frac{\sum_{i=1}^{m} I(X = x_j | Y = C_k)}{\sum_{i=1}^{m} I(Y = C_k)}, k = 1, 2, ..., K, \text{is the number of occurrence of feature attribute } x_j \text{ in sample category } C_k.\]

According to the assumption that the characteristic attributes are independent of each other, and then according to Bayes' theorem, the posterior probability of each category is

\[
P(Y = C_k | X = x_j) = \frac{P(X = x_j | Y = C_k)P(Y = C_k)}{P(X = x_j)}.
\]  

In the posterior probability of each category, it only need to obtain the maximum value of the molecular part. Therefore, the final formula is

\[
C = \text{argmax} P(Y = C_k) \prod_{j=1}^{n} P(X = x_j | Y = C_k)
\]  

There are two main advantages of Naive Bayes algorithm. One is that it is a simple algorithm, with fast speed and good effect, and the other is that it has high stable classification efficiency because of its strong theoretical basis as support, but its biggest flaw is that assumption which is almost impossible in reality. Therefore, when a data set with strong correlation between the data appears, the effect of Naive Bayes algorithm is not satisfactory.
2.2. KNN Nearest Neighbor

The core idea of the KNN nearest neighbor algorithm is that if most data in a sample belong to a certain category in K nearest neighbor samples in the feature space, then the sample also belongs to this category, and has the sample characteristics of this category. The algorithm is mainly used in text classification, pattern recognition, cluster analysis, and multi-classification fields.

Step 1: To determine the vector representation of the test text based on the feature words for the new test text.

Step 2: To calculate the similarity between the text and each text in the training text set, the formula is:

\[
\text{Sim} (d_i, d_j) = \frac{\sum_{k=1}^{M} w_k \times w_{jk}}{\sqrt{\sum_{k=1}^{M} w_k^2 \times \sum_{k=1}^{M} w_{jk}^2}}
\]  

(5)

d_i is the feature vector of the test text, d_j is the center vector of category j , M is the dimension of the feature vector, W_k is dimension k of the vector.

According to the calculated text similarity, it is essential to find the K texts that are most similar to the new test text in the training text set. The value of K is generally determined by using an initial value, and then adjusted according to the results of experimental tests. The range of the initial value is generally hundreds to thousands.

Step 3: In the K nearest neighbors of the test text, to calculate the weight of each category in turn, the formula is:

\[
P(X, C_j) = \frac{1}{\sum_{\text{other}} \text{Sim}(x, d_i) y(d_i, C_j) - b \geq 0}
\]

(6)

x is the feature vector of the test text. Sim (x, d_i) is the calculated similarity. b is the threshold. y (d_i, C_j) takes the value 0 or 1. If d_i belongs to C_j, the function value is 1, otherwise, it is 0.

Step 4: To compare the weights of the categories and divide the text into the category with the largest weight.

The advantage of the KNN nearest neighbor algorithm is that it does not require any input data to make any assumptions. It achieves a high rate of accuracy on dealing with the relatively rare data sets or the data sets with multiple categories. However, the KNN nearest neighbor algorithm is limited to the balance and scale of the data set. When the distribution of the data set is unbalanced, the classification effect is not ideal. When the size of the data set is very large, the dimension of the feature vector increasing accordingly, it leads to higher calculation complexity and relatively high cost of time and space.

2.3. Support Vector Machines

Support vector machines (SVM) are mainly used to solve the problem of data classification. It is achieved by constructing a classification function or classification model. The algorithm flow is as follows.

- Start
- Choose training set
- Construct an optimization problem and solve it
- Calculate parameters, \( \text{K} \)
- Obtain the result of hyperplane-assistance function
- End

Figure 3. SVM algorithm flow

The algorithm has a very wide range of applications in the field of pattern recognition, such as character recognition, face recognition, pedestrian detection, text classification, image filtering and classification, bio-informatics and many other aspects.

The principle of support vector machines can be summarized as follows.
There is a given training set, \( T = \{ (x_1, y_1), (x_2, y_2), \ldots, (x_i, y_i) \} \), \( x_i \in \mathbb{R}^n \), \( y_i \in \{1, -1\} \), \( i = 1, 2, \ldots, n \). SVM finds an optimal classification hyper-plane (or discriminant function) that meets the classification requirements. This hyper-plane can not only ensure high classification accuracy, but also make the blank area on the hyper-planes sides reach the maximum value (as shown in the figure below).

![Hyper-plane in two-dimensional space](image)

**Figure 4.** Hyper-plane in two-dimensional space

The hyper-plane is referred to as \( (w^T x) + b = 0 \). Since the blank area on both sides of the hyper-plane should be as large as possible, in other words, the distance between the two types of samples to the hyper-plane should be as large as possible, we need to add constraints to \( w \) and \( b \). According to the formula of the distance from the point to the plane, the distance from each sample point to the hyper-plane is

\[
d_i = \frac{|w^T x_i + b|}{||w||}
\]

The hyper-plane is

\[
\min_{x_i} |w^T x_i + b| = 1
\]

Finally, the distance from two types of samples to the hyper-plane is:

\[
d(w,b) = \min_{x_i, y_i = -1} |w^T x_i + b| + \min_{x_i, y_i = 1} |w^T x_i + b| = \frac{2}{||w||}
\]

The maximum value of the above formula can be obtained by the minimum value of \( \frac{1}{2} ||w||^2 \), and this minimum value can be solved by the Lagrange method.

To construct the Lagrangian function and derive the parameters \( w, b, \alpha \).

\[
L(w, b, \alpha) = \frac{1}{2} ||w||^2 - \sum_{i=1}^{n} \alpha_i y_i (w^T x_i + b) + \sum_{i=1}^{n} \alpha_i
\]

\[
\frac{\partial}{\partial w} L(w, b, \alpha) = w - \sum_{i=1}^{n} \alpha_i y_i x_i \quad \frac{\partial}{\partial b} L(w, b, \alpha) = -\sum_{i=1}^{n} \alpha_i \quad \frac{\partial}{\partial \alpha} L(w, b, \alpha) = -\sum_{i=1}^{n} y_i (w^T x_i + b) + n
\]

Due to the duality of Lagrangian function \( \min_{w,b} \max_{\alpha} L(w, b, \alpha) \Rightarrow \max_{\alpha} \min_{w,b} L(w, b, \alpha) \), the first step is to find the minimum value.

\[
\frac{\partial}{\partial w} L(w, b, \alpha) = 0 \Rightarrow w^* = \sum_{i=1}^{n} \alpha_i y_i x_i \quad \frac{\partial}{\partial b} L(w, b, \alpha) = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0
\]

The formula after bringing the above formula into the Lagrangian function is:

\[
L(w, b, \alpha) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i x_j) - \sum_{i=1}^{n} \alpha_i y_i \sum_{j=1}^{n} \alpha_j y_j x_i x_j + \sum_{i=1}^{n} \alpha_i = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i x_j) + \sum_{i=1}^{n} \alpha_i
\]

At this time, the solution problem can be converted.

\[
\max_{\alpha} L(w, b, \alpha) = -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i x_j) + \sum_{i=1}^{n} \alpha_i \Rightarrow \min_{\alpha} L(w, b, \alpha) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i x_j) - \sum_{i=1}^{n} \alpha_i
\]
The decision function is \[ f(x) = \text{sign}(w^* \cdot x + b) = \text{sign}(\sum_{i=1}^{N} \alpha_i y_i x_i \cdot x - x_i \cdot \sum_{i=1}^{N} \alpha_i y_i x_i) \].

The main advantage of the support vector machine algorithm is that it is robust. Its decision function is determined by several support vectors, so the calculation complexity only depends on these support vectors which avoids the problem of high dimension and making it difficult to run. However, it also has some restrictions on the target data set. When the scale is large, the efficiency will be unsatisfactory. In addition, the traditional SVM algorithm can only solve the problem of classification. In an era of complicated data, solving multi-category problems is unavoidable.

3. The Application of Machine Learning in Text Classifier

3.1. Evaluation of Classifier

Accuracy is the most common evaluation index. Regardless of category, as long as the prediction is correct, the number of numerators is counted, and the denominator is the number of all samples, so the formula is Accuracy = number of samples predicted correctly / total samples.

Precision refers to the number of samples that are positive in the class determined by the classifier. The formula is Precision = predicted to be positive Number of texts / Actual number of positive texts.

Recall refers to how many classifiers have identified or judged the correct number in all positive examples. The formula is Recall = number of texts predicted to be positive / actual number of texts that should be positive.

The F1 value is based on the reconciled average value of the precision and recall rates to allow both to reach the highest point at the same time because the precision rate and recall rate are a pair of conflicting standards. The larger the F1 value is, the better the classification performance the classifier has. The formula is \[ F_1 = \frac{(2 \times \text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})} \].

4. The Experiment

The experimental data is a set of spam texts, which are divided into two categories: normal text messages and spam text messages, which contain more than 10,000 texts. 1,000 normal texts and 1,000 spam texts are extracted and divided into a training set and a test set at a ratio of 8:2, that is, the training set has 1,600 texts and the test set has 400 texts.

4.1. Background of the Experiment

Spam text messages refer to text messages that are sent to users without the user's consent and permission, and the text messages are information that the user does not want to receive or the user cannot refuse to receive according to their own wishes.

4.2. Detail Test

First of all is to pre-process the experimental data. The text pre-processing process in Chinese and English text is roughly the same, divided into five steps, as shown in the figure.

![Figure 5. Hyper-plane](image)

We use the Naive Bayes algorithm, KNN nearest neighbor algorithm and support vector machine algorithm to learn the classification model on the training set text, and then use the trained classification model to perform classification prediction on the test set text. The experimental results are shown below.
Table 1. The Results

|                 | Accuracy | Precision | Recall | F1   |
|-----------------|----------|-----------|--------|------|
| Naive Bayes     | 90%      | 87%       | 93%    | 90%  |
| KNN             | 55.75%   | 55%       | 44%    | 49%  |
| SVM             | 91.25%   | 100%      | 82%    | 90%  |

5. Conclusion

The amount of data in today's era is growing at a geometric level. People's demand for the conversion of massive data into useful information is very urgent. Machine learning needs to target large-scale and complex data sets, especially in large-scale complex applications such as Internet content. The performance bottlenecks encountered are these key research objects and major breakthroughs. If machine learning has further major development in data mining, it will inevitably bring greater benefits to the world.

Acknowledgement

To acknowledge all of the people who have contributed to this paper in some manner.
References
[1] Peng Ge. A Summary of Research on Spam Filtering Based on Naive Bayes Algorithm [J]. Computer knowledge and technology, 2020, 14: 250-253.
[2] Li Jingmei. A naive Bayes classifier in text processing [J]. Modern Electronic Technology, 2005, 28 (14): 40-41.
[3] Zhuo Chen. The Lao Text Classification Method Based on KNN[J]. Procedia Computer Science, 2020, 166: 523-538.
[4] Huang Chao. Chinese text classification based on improved K nearest neighbor algorithm[J]. Journal of Shanghai Normal University (Natural Science Edition), 2019, 01: 102-107.
[5] Liu Xiaoliang. Applicability of SVM for text classification [J]. Computer Engineering and Science, 2010, 32 (6): 106-108.
[6] Xu Juan. Research on Chinese Spam Prediction System Based on SVM[J]. Digital Technology and Application, 2020, 01: 48-49.
[7] He Qing. Overview of machine learning algorithms under big data[J]. Pattern recognition and artificial intelligence, 2014, 27(4): 327-336.