Research on Text Classification Algorithm Based on Machine Learning

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

Abstract. In the era of big data, the data we analyze is no longer limited to numbers, while even more data analysis is based on text. Facing massive and complex data, traditional analysis methods cannot effectively process text data. Manual processing of text data not only consumes a lot of time and manpower, but also it is difficult to ensure the fairness and objectivity of the assessment. The development of machine learning provides us with an effective method of text mining. This article will introduce three classic algorithms in machine learning for elaboration, and apply them into a specific experiment to show the wisdom of machine learning in the field of text classification.

1. Introduction
Text mining refers to mine the potentially valuable information in the text. There are three differences between text data and numeric data. The first is that text data is unstructured. The second is that it is closely related to human language and mind. The last one is the meaning of the text is not unique. Machine learning is an interdisciplinary subject that involves many fields and is closely related to many subjects, including probability theory, statistics, computer science, and so on. It has been successfully applied to solve the problems of artificial intelligence. It has the ability to learn from the data and make predictions on the data. Because of this characteristic, it does not depend on static program instructions. Through the input of a sample, it builds a model and then it is operated as the basis of prediction or final decision. That is the reason why machine learning can efficiently perform text mining.

2. Algorithms of Machine Learning
This paper introduces three typical text classification algorithms, all of which have good classification effect.

2.1. Naive Bayes Algorithm
Naive Bayes algorithm is one of the few classification algorithms based on probability theory. It has been successfully applied in 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.
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.

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 \( C_k \).

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

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

In the posterior probability of each category, it only need to obtain the maximum value of the following formula.

\[
P(X = x_i | Y = C_k) = \arg\max_{C_k} P(Y = C_k) \prod_{j=1}^{n} P(X = x_j | Y = C_k)
\]

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 it 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

KNN Nearest Neighbor algorithm is a basic classification method, which takes the feature vector of the actual example as input, calculates the distance between the new sample and the feature value of the training sample set, and then selects K nearest neighbor is classified and judged.
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:

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

$$d_i$$ 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:

$$P(X, C_j) = \frac{1}{\sum_{\text{other}}^{\text{kNN}} \sum_{y \in \text{C}_j} \text{Sim}(x, d_i)y(d_i, x) - b \geq 0}$$

$$x$$ is the feature vector of the test text. $$\text{Sim}(x, d_i)$$ 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 data items in the database are mapped to a certain category of the specified category, so as to predict the unknown category. The algorithm flow is as follows.

![Figure 2. SVM algorithm flow](image)

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 machine can be summarized as followed. 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).
Figure 3. Hyper-plane in two-dimensional space

The hyper-plane is referred to as \((w^*x)+b=0\). 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

\[
d = \frac{|w^T x + b|}{||w||}
\]

hyper-plane is \(d\), \(x\) is a function variable, \(w\) is a normal vector. Therefore, the constraints

\[
\min |w^T x_i + b| = 1
\]

of the hyper-plane is \(x\).

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

\[
d(w,b) = \min_{i,x_i=1} \left( \frac{|w^T x + b|}{||w||} \right) + \min_{j,x_j=-1} \left( \frac{|w^T x + b|}{||w||} \right) = \frac{1}{||w||} \left( \min_{i,x_i=1} |w^T x_i + b| + \min_{j,x_j=-1} |w^T x_j + b| \right) = \frac{2}{||w||^2}
\]

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 \(w\), \(b\), \(\alpha\).

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

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

Due to the duality of Lagrangian function \(\frac{\partial}{\partial \alpha} L(w,b,\alpha) \Rightarrow \max\min_{\alpha} L(w,b,\alpha)\), 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:

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

At this time, the solution problem can be converted.

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

\[
f(x) = sign(w^* \cdot x + b) = \sum_{i=1}^{n} \alpha_i y_i x_i \cdot x + y_i \cdot x_i - \sum_{i=1}^{n} \alpha_i y_i x_i
\]

The decision function is
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. The formula is Accuracy = number of samples predicted correctly / total samples. This indicator ignores the specific category, so when the sample is not distributed in balance, the indicator will be invalid. In actual situations, the accuracy rate is usually only used as a preliminary reference indicator.

Precision refers to whether the samples determined by the classifier as positive examples are really positive examples. The formula is precision = Number of texts correctly predicted to be positive / Number of texts predicted to be positive.

Recall refers to whether to find all the correct samples in the full positive examples. The formula is recall rate = Number of texts correctly predicted to be positive / Actual number of texts that should be positive.

3.2. The Experiment

The experimental data contains more than 10,000 spam texts, consisting of two types: normal text messages and spam text messages. At first, 1,000 normal texts and 1,000 spam texts are extracted in random. Then, In order to train the model, it is necessary to extract 800 normal texts and 800 spam texts. The remaining data set is used as a test set to evaluate the model.

3.2.1. Detail test. We apply Naive Bayes algorithm, KNN nearest neighbor algorithm and support vector machine algorithm to conduct the experiment. We now analyze the results one by one.

1) From the view of accuracy, the effect of support vector machine classifier is slightly better than that of Naive Bayes classifier. Both are far better than the classification effect of KNN classifier.

2) In the sight of precision, SVM classifier have the highest classification degree of precision.
3) In the respect of recall, Naive Bayes classifiers do the best job in finding full samples.

![Figure 6. The results of recall](image)

3.2.2 Results. On the whole, Naive Bayes classifier and SVM classifier have good effect on text classification. The classification effect of the nearest neighbor classifier is not very satisfactory.

![Figure 7. The results](image)

4. Conclusion

Experimental results show that Naive Bayes classifier and SVM classifier have achieved good results in spam messages recognition. However, with the increasing data volume of text mining in the era of big data, there are still many problems worthy of study in machine learning, such as category scale and performance bottlenecks in large-scale complex Internet applications, which will become the focus and main breakthrough direction of text classification related research. Therefore, deeper research should be conducted on the inherent defects of machine learning in the future to overcome the defects, so that it can solve the problems of the specific text mining and classification application.

References

[1] Zou Xiaohui. Application of Naive Bayes algorithm in text classification[J]. Digital Technology and Application (Algorithm Analysis), 2017: 132-133.
[2] Huan Zhang. Class-specific attribute value weighting for Naive Bayes[J]. Information Sciences, 2020, 508: 260-274.
[3] Zhang Ning. Text classification using KNN algorithm[J]. Computer Engineering, 2005, 31 (8): 171-172, 185.
[4] Zhuo Chen. The Lao Text Classification Method Based on KNN[J]. Procedia Computer Science, 2020, 166: 523-538.
[5] Liu Xiaoliang. Applicability of SVM for text classification[J]. Computer Engineering and Science, 2010, 32 (6): 106-108.
[6] Qi Hengnian. Summary of support vector machine and its application research[J]. Computer Engineering, 2004, 30 (10): 6-8.

[7] Ding Shifei. A Summary of Support Vector Machine Theory and Algorithm Research[J]. Journal of University of Electronic Science and Technology of China, 2011, 40 (1): 2-10.

[8] He Qing. Overview of machine learning algorithms under big data[J]. Pattern recognition and artificial intelligence, 2014, 27(4): 327-336.