Advances in Intelligent Systems Research, volume 147

International Conference on Network, Communication, Computer Engineering (NCCE 2018)

Improvement and Implementation of Feature Weighting Algorithm TF-IDF in Text Classification

Weisi Dai [a)]

Department of Informatics, Beijing University of Technology, Beijing 100124, China.

[a)] Corresponding author: 962841737@qq.com

Abstract. Based on the introduction of the traditional feature weighting algorithm TF-IDF, based on the phenomenon that the eigenvalue extraction is not effective when the text to be classified is not uniform, an improved TF-IDF algorithm is proposed in this paper, which considers the uneven text distribution Inside. The experimental results show that the results obtained by the classification algorithm using the improved algorithm are better than the original algorithm in terms of accuracy and recall and make up for the defects of the original TF-IDF algorithm.

Key words: Classification algorithm, TF-IDF

INTRODUCTION

With the continuous development and application of information technology, the improvement of storage technology and the rise of technologies such as social networking websites and mobile internet, the amount of information in the world is rapidly increasing. At present, the number of texts that are updated on the Internet every day in the world is above ten billion. The content covers almost every aspect of society. On the one hand, the more information we can get through the Internet, the more it becomes more and more laborious to quickly and accurately classify the specific information that individuals need in such a huge amount of information. Therefore, how to find the specific information that individuals need accurately, rapidly and comprehensively in the large-scale textual e-commerce information has become a very important research.

The main function of text categorization is to classify and assign the related document collection. The main basis of the text categorization is to gather the training documents that have been pre-assigned as tags in the text training process. In the face of a large number of documents, the corresponding text features are also quite high. Therefore, in the face of high-dimensional text features, the dimensionality disaster will be greatly affected if the dimensionality reduction is not performed. Therefore, feature reduction is an important part of the text classification process. Feature extraction and feature weighting are two major categories of feature dimensionality reduction and have been widely used in the field of text categorization. After dimensionality reduction of the text, the feature set that can best represent the text category information can be obtained. However, not all feature words in the collection have the same effect on the result of the final text classification. Some feature words contain more category information, indicating that they are more capable of distinguishing textual analogies, while some word-containing categories with little information, its ability to distinguish between text categories is weak. Therefore, the feature weighting process needs to be performed next.

Feature weight refers to the process of calculating the weight of the category information for each feature item in the feature set according to the specified weighting algorithm. Through feature weighting, the spatial distribution of text feature sets can be effectively improved, and the intra-class concentration and inter-class scatter of text spaces can be increased, so that the text category information can be more rationalized and the accuracy of the text classification algorithm can be improved.

At present, the most commonly used feature weighting methods are: TF-IDF, square root function, Boolean function and so on. The TF-IDF algorithm is widely used in text categorization due to its advantages of fastness, simplicity and accuracy.

In this paper, the TF-IDF algorithm is introduced in detail, and an improved algorithm is proposed to deal with the poor performance of the original TF-IDF when the classified texts are unevenly distributed. The effectiveness of the algorithm is verified through experiments.

TF-IDF ALGORITHM

TF-IDF algorithm understood as two parts, respectively, the characteristic frequency (TF) and inverse document frequency (IDF), calculated in two parts were calculated and then multiplied by the respective results obtained TF-IDF algorithm results.

Feature Frequency (TF), the number of times feature items appear in the document. Generally, the features with higher frequency appear more weight in this kind of text. If more than one feature appears in the text of a category, it indicates that the feature is more important to the text. The feature frequency weight calculation formula for the feature tk:

$$w_{tk} = tf_{ik} \tag{1}$$

Where tf_{tk} represents the number of times tk appears in the text.After calculation, each text has different eigenvalues, and the weights are also different. Compared with other feature-weighted algorithms, such as the Boolean function weighs only 1 and 0, there is a good result. However, TF does not consider the length of the document will have a certain effect on the frequency of tk in the document. Due to little or no positive information, this subset of features would have contributed less to the classification but may have greater weight as the text grows in length, which may affect the classification results.

Merely considering the word frequency does not adequately represent the importance of the word to the text. Many stop words, such as articles, pronouns, and auxiliary words, appearing in texts interfere with the calculation of feature weights, because they occur more frequently in all texts, but the actual classification contributes less. If we can reduce the side effects of high-frequency words that exist in most texts, we need to introduce the inverse document frequency (IDF), which uses the number of texts containing features as parameters to construct feature weights.

The basic idea of IDF is that if a feature appears in a document, but at the same time the feature also appears in many documents, then the feature is less important to the classification. Feature tk's inverse document frequency IDF (tk) in the entire text collection is calculated as:

$$IDF(t_k) = \log_2(N / n_k) \tag{2}$$

Where nk is the number of texts containing feature tk and N is the total number of texts in the text set.

Therefore, the basic idea of TF-IDF is that if the frequency of the feature tk appearing in the entire text set is lower and the frequency of appearance in a certain type of text is higher, then the contribution of tk to the classification should be given a higher weight. The weight of the characteristic tk in the text is calculated as:

$$w_{ik} = tf_{it} \times \log_2(N / n_k) \tag{3}$$

In general, the feature weights need to be normalized to eliminate the effect of text length on feature weights. Normalized formula:

$$w'_{ik} = \frac{tf_{ik} \times \log_2(N / n_k)}{\sqrt{\sum_{j=1}^{M} [tf_{ik} \times \log_2(N / n_k)]^2}} \tag{4}$$

IMPROVED TF-IDF ALGORITHM

Defect of the TF-IDF Algorithm

TF-IDF treats the document set as a whole, especially the calculation of IDF. There are obvious defects in the text classification: In the formula of IDF, let nk = n1 + n2, n1 denote the number of documents with feature word m in ci, n2 Indicates the number of documents containing feature word m in other classes. When n1 >> n2, the value of IDF is small when the total number of documents N is constant. However, the actual situation is that the eigenvalue m appears in the class ci much more frequently than in other classes, and the characteristic word m should have a good ability to discriminate, but here it is the opposite of the expected result.

TABLE 1. M1 M2 distribution characteristics

Category	M1	M2
C1	9	5
C2	1	5

The above table as an example, there are C1, C2 two categories, m1, m2 two eigenvalues. The number of documents containing m1 and m2 feature words in C1 and C2 were 9 articles, 5 articles, 1 article, and 5 articles respectively. ID1 = log (10/9) = 0.0496, and IDF2 = log (10/5) = 0.303 when the IDF values of m1 and m2 in class C1 are IDF1 and IDF2, respectively. The values of IDF1 and IDF2 indicate that m2 has a better classification effect than m1. However, according to actual observations, the m1 distribution is uneven and the m2 distribution is uniform, indicating that m1 has better class discrimination ability than m2.

Improve Algorithm

Based on the above-mentioned drawbacks, a relative frequency is used instead of the traditional IDF formula when calculating the IDF part. The target class to be classified is defined as a positive class, while the other non-target classes are defined as opposite classes. There are two cases for a document: 1. The document contains the feature tk and belongs to the class; 2. The document contains the feature tk and belongs to the reverse class. Assume that case 1 is a and case 2 is b.

Assuming that the total value of a+b is fixed, regardless of whether a>>b, a=b, or a<<b, the traditional TF-IDF calculation is equivalent to the three cases. However, the actual analysis by the previous section shows that in fact the distinction between the three cases is different. Therefore, a new algorithm is proposed. When the proportion of a is greater than the proportion of b, the stronger the ability of the feature word to distinguish the positive and negative classes, the higher the weight assigned to it. This results in a new algorithm calculation formula:

$$idf = \log_2(2 + \frac{a}{b}) \tag{5}$$

To make the value of the relevant frequency always positive, the constant 2 is added to the formula. As a result, the above formula is modified as:

$$idf = \log_2(2 + \frac{a}{\max(b,1)}) \tag{6}$$

According to this formula, calculate the data in the table in the previous section and set the target class to C1, $idf1 = \log_2(2 + \frac{9}{1}) = 3.46$, $idf2 = \log_2(2 + \frac{5}{5}) = 1.58$.It shows that the classification effect of m1 is better than m2, which is consistent with the facts.

The final weight is calculated as:

$$w_{ik} = tf_{ik} \times \log_2(2 + \frac{a}{\max(b,1)}) \tag{7}$$

EXPERIMENT

The experimental environment for this experiment was Matlab R2010a. The experiment will use the KNN classifier to measure the original feature weighted TF-IDF algorithm and the improved algorithm. The data set used in the experiment is 20-Newsgroups. This data set is an international standard data set with a total of more than 20,000 newsgroup documents and is divided into 20 groups of topics. The topics are well dispersed. Experiments will compare the effect of TF-IDF and its improved algorithm on the performance of the classifier, and at the same time verify through experiments whether the improved algorithm proposed in this paper further improves the classifier performance. The experimental results are shown in Fig.1 as follows:

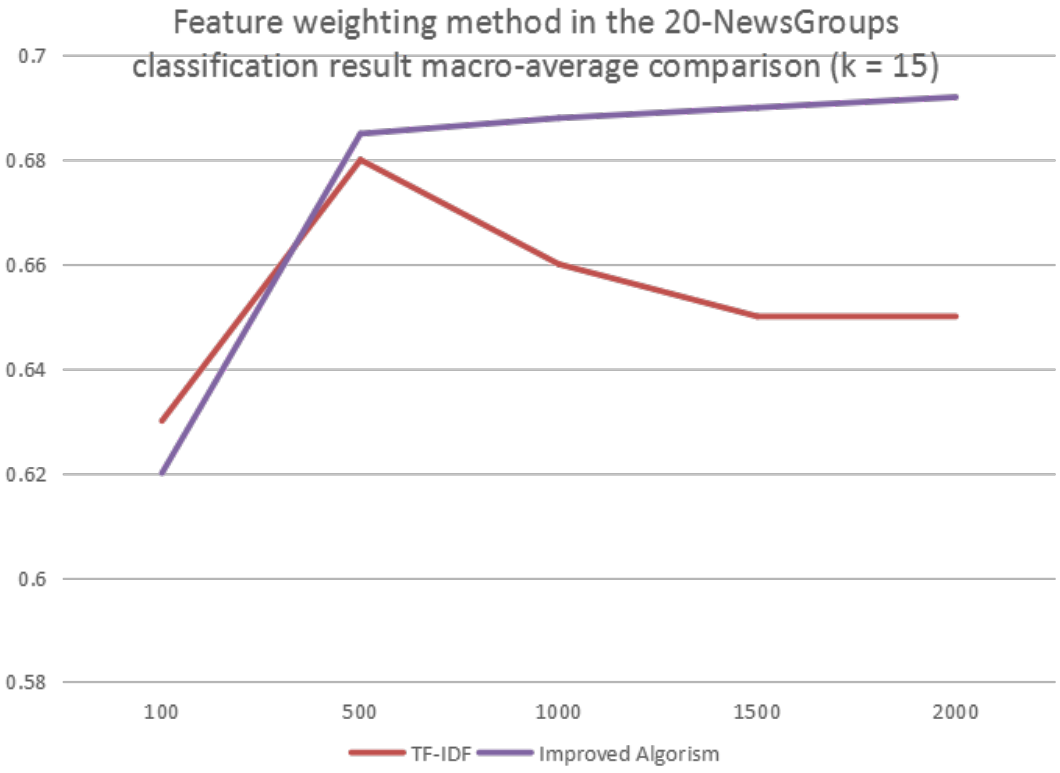

FIGURE 1. Feature weighting method in the 20-NewsGroups classification result

It can be seen from the experimental data that when the number of features is less than 500, the TF-IDF algorithm is similar to the macro average obtained by the improved algorithm, but when the number of features exceeds 500, the result of feature weighting using the TF-IDF algorithm starts to decrease. , But the result of feature extraction using improved algorithm is still increasing, which shows the effectiveness of the improved algorithm.

CONCLUSION

This paper introduces and analyzes the TF-IDF feature weighting algorithm and points out that when the text to be classified is distributed unevenly, the eigenvalue extraction is not effective. In this paper, an improved TF-IDF algorithm is proposed, compared with the document with the eigenvalue in the original formula, the ratio of the document to the total eigenvalue can be replaced by the positive and negative documents, so that the algorithm can

better deal with the confusion when the uneven distribution. The experiment proves the effectiveness of this algorithm.

REFERENCES

1. An Evolutionary Approach for Image Retrieval Based on Lateral Inhibition[J]. Bai Li. Optik - International Journal for Light and Elect. 2016
2. Single-beam phase retrieval with partially coherent light illumination[J]. Meiling Zhou,Junwei Min,Peng Gao, Yansheng Liang,Ming Lei,Baoli Yao. Journal of Optics. 2016 (1)
3. Optical image hiding using double-phase retrieval algorithm based on nonlinear cryptosystem under vortex beam illumination[J]. Xiaogang Wang,Wen Chen,Xudong Chen. Journal of Optics. 2015 (3)
4. Probabilistic models in IR and their relationships[J]. Robin Aly,Thomas Demeester, Stephen Robertson. Information Retrieval. 2014
5. Latent word context model for information retrieval[J]. Bernard Brosseau-Villeneuve,Jian-Yun Nie,Noriko Kando. Information Retrieval. 2014 (1)
6. Chen G B. A Distributed Dynamic Load Balancing Scheduling Algorithm[J]. Journal of Guangxi Teachers Education University, 2014.
7. Godfrey B, Lakshminarayanan K, Surana S, et al. Load balancing in dynamic structured P2P systems[C]// Joint Conference of the IEEE Computer and Communications Societies. IEEE, 2004:2253-2262 vol.4.