Research on TFIDF Algorithm Based on Weighting of Distribution Factors

Xinming Zhang, Yuanbo Shi and Haiping Wei
School of Computer and Communication Engineering, Liaoning Shihua University, Liaoning, China
Email: zxm_9603@163.com; 329265562@qq.com; 416089317@qq.com

Abstract. The current TFIDF (Term Frequency and Inverted Document Frequency) algorithm cannot effectively reflect the relationship between the importance of a word and its distribution. This paper proposes a Class Variance-Term Frequency and Inverted Document Frequency algorithm. This algorithm improves the TFIDF algorithm based on three distribution factors: category, inter-class and variance. In order to measure the optimization effect of this method, three algorithms were compared using the original algorithm, improved algorithm and TFIDF algorithm based on dual parallel calculation model. Experiments show that the improved algorithm has significantly improved recall, accuracy, and F metric values comparing with the original algorithm, and has improved compared with the TFIDF algorithm based on dual parallel calculation model. Therefore, the improved algorithm can well adapt to the demand for feature word extraction and has better text classification performance.

Keywords. Tfidf; unbalanced data set; variance; text classification.

1. Introduction
With the popularity of the network, massive amounts of text information have been produced on the network. To meet the diversified needs of users in the context of massive data, text data needs to be effectively classified. Text classification technology plays an important role in information retrieval. Its main task is to determine its category based on the text content under a predetermined set of category tags [1-2]. Text categorization usually transforms text information into eigenvectors. Because any word in the data set can become a keyword, it will cause the final feature vector dimension to be too high and affect the classification result. A natural way to reduce the dimension of feature space is feature extraction [3].

TFIDF (Term Frequency and Inverted Document Frequency) algorithm is a more commonly used method for text feature extraction. There are many improved methods for TFIDF algorithm [4]. Ref. [5] aimed at the community question answering system, after classifying the question sentence according to the user’s query intent, and adjusting the weights according to the distribution of feature words in the category. Ref. [6] proposed a feature weighting algorithm based on information entropy theory considering the rate of keyword in the document and the concentration of the feature words in the training set and the degree of dispersion in each category. Ref. [7] proposed an improved TFIDF algorithm that uses information entropy and relative entropy in information theory as calculation factors. Ref. [8] proposed a new distance-based feature word weighting method. This algorithm performs more prominently in news classification and clustering. The above algorithms all take into account the optimization calculation of the feature words in the classified text set with respect to the categories, but
in an imbalanced data set with a large number of words in a single text, the above algorithms cannot effectively calculate accurate weights.

Aiming at the text feature extraction of an imbalanced data set with a large number of words in a single text, this paper proposes a TFIDF-CV (Class Variance-Term Frequency and Inverted Document Frequency) algorithm. The algorithm takes into account the distribution of feature word in a single text, and the weight of feature words changes according to the distribution of feature words. For example, feature words that are evenly distributed in an article have a higher weight than feature words that are concentrated in a paragraph. In addition, the algorithm also defines a category distribution factor to weaken the weight of keywords with a large number of texts in different categories in the text set and increase the weight of keywords with few texts in different categories.

2. Traditional TFIDF Algorithm

TFIDF is an algorithm to calculate the weight of feature words according to the characteristics of words. It combines TF (word frequency) and IDF (document inversion frequency). The central content is that the higher the frequency of the word in the text, the higher the weight, and the smaller the number of texts containing the word in the text set, the higher the weight of the word. The calculation formula is as follows [9-10]:

$$W_{ij} = tf_{i,j} \times idf_i = \frac{n_{i,j}}{\sum_t n_{i,t}} \times \log \frac{N}{n}$$  \hspace{1cm} (1)

Among them, $W_{ij}(i=1,2,\cdots,n)$ is the weight of the feature word $t_i$ in the text $d_j$, $tf_{i,j}$ is the frequency of the feature word $t_i$ in the text $d_j$, and $N$ is the text corpus. The total quantity of Chinese texts, $n$ is the quantity of texts containing the feature word $t_i$ in the text corpus. In order to limit the final result to 0-1, the algorithm needs to be normalized [11]. The formula is as follows:

$$W_j = \frac{\sqrt{\sum_{i=1}^{T} \left( tf_{i,j} \times \log \frac{N}{n+1} \right)^2}}{\sum_{i=1}^{T} \left( tf_{i,j} \times \log \frac{N}{n+1} \right)}$$  \hspace{1cm} (2)

The advantages of TFIDF algorithm are that the algorithm is easy to understand and has a high accuracy rate in large task classification, so it has been widely used. However, in the class with a small number of documents in the data set, when the feature words mainly exist in this class, it should be given high weight to reflect the text features, but it was abandoned because the number of documents was too small in total. In the classification text set, whether in different categories, within a single category, or in a text file, the algorithm does not consider the distribution of feature words in the text. For example: between different categories, If a word appears in one category much more often than in the other, the feature weight of this feature word is obviously high and cannot be reflected in the algorithm. Also in a single text file, the value of feature words concentrated in a certain part and scattered throughout the article is obviously different, but the calculation results are the same.

3. Improvement of TFIDF Based on Word Frequency Distribution

In imbalanced data sets with category distributions, the traditional feature selection algorithm usually tends to choose words in categories that have a large amount of text. Meanwhile, the difference in the distribution of word between categories and word frequency in a single text will lead to different weights of the final feature. Therefore, this paper proposes three distribution factors based on the different distribution modes of words in the data set to improve the defects of the original algorithm.

3.1. Category Distribution Factor

This factor mirrors the distribution of different texts in the different classifications. This factor is aimed at the situation that the number of texts of each class in the different classes of feature words accounts for the amount of texts of all classes, the purpose of which is to alleviate the limitation of inverse
document frequency calculation weight tendency of large categories to ignore small categories. It can be calculated by the total number of documents \( N \) and the number of documents \( n_i \) in different categories \( C_i \) according to the given formula 3. When the total number of texts in category \( C_i \) accounts for a small part of the total number of documents \( N \), it indicates that the number of \( C_i \) documents in this class is small and belongs to a small class. In order to avoid the limitation that traditional algorithms tend to ignore large categories, large categories and small categories are balanced by finding the quotient of the total number of documents and the number of class \( C_i \) documents. The formula for class distribution factor \( \alpha \) is:

\[
\alpha = \log_2 \left( \frac{N}{n_i} \right)
\]  

(3)

3.2. Interclass Distribution Factor

The distribution factor within a category shows the relationship between the distribution of words in different texts and the weight of words in a category. This factor considers the distribution of feature words in all classes, and makes up for the limitation that the inverse document frequency cannot handle multiple classes. It can use the number of texts containing a certain word in the text set \( t_{2ki} \) and the number of all texts containing the word in different categories \( t_{ki} \) to calculate by formula 4. The importance of the distribution of feature words to the class is reflected by finding the proportion of feature words between the classes. When the number of texts of feature words in the class \( C_i t_{2ki} \) occupies the main part of all texts containing the feature word \( t_k n_{2tk} \), it indicates that the feature words are mainly distributed in a class, which further indicates that Feature words can distinguish different categories well and should be given higher weight. The formula for calculating the distribution factor \( \beta \) between classes is:

\[
\beta = \log_2 \left( 2 + \frac{t_{2ki}^2}{n_{2tk}} \right)
\]  

(4)

3.3. Variance Distribution Factor

This factor is based on the distribution of feature words in a text, which is a further improvement of the word frequency and is reflected by the variance. Variance [12] can measure the degree of dispersion of a set of data. The distribution of a feature word in a document can be represented by the variance. The larger the variance, the more dispersed the feature words are in a document, and the more dispersed the feature, the stronger the ability to distinguish documents. The smaller the variance, the more centralized the words are in a document, and the more centralized the distribution, indicating that the feature words are mainly concentrated in a section or a paragraph, and the ability to distinguish document categories is weak. The variance calculation formula is:

\[
\sigma^2 = \frac{\sum_{i=1}^{n3} (t_{3ki} - \mu)^2}{n3}
\]  

(5)

where \( \sigma^2 \) represents the overall variance, \( t_{3ki} \) represents the position of the feature word \( t_k \) in the document (in digital form), and \( \mu \) represents the overall mean (the calculation method of \( \mu \) is to find the average of the total value of the appearance positions of each feature word Value), \( n3 \) represents the total number of feature words \( t_k \) appearing in the document. In order to avoid that the variance value is too large to affect the weight calculation, taking into account that a feature word has multiple variance values in multiple texts, the following processing is performed:

\[
\gamma = \log_2 \left( 2 + \sum_{j=1}^{m} \sigma_j^2 \right)
\]  

(6)
where $\gamma$ represents the variance distribution factor, and $j$ represents the $j$-th text of the same kind, $\sigma_j^2(j = 1, 2, \cdots, m)$ represents the variance of the feature word in the first text.

3.4. Improved TFIDF Algorithm

This paper proposes three distribution factors based on the different distribution modes of words in the data set. The three distribution factors are independent of each other and collectively support the weight measurement of an imbalanced data set with a large number of words in a single text. At the same time, the value of each factor has a positive correlation with the weight, so the final value is calculated as a product.

Based on the above equations (1), (3), (4), (6), the improved weight formula TFIDF-CV is

$$TFIDF-CV = tf_{i,j} \times idf_i \times \alpha \times \beta \times \gamma$$

$$= \frac{n_{i,j}}{\sum n_{k,j}} \times \log \frac{N}{n} \times \log_2 \left( \frac{N}{n t_i} \right) \times$$

$$\log_2 \left( 2 + \frac{t_{2h}}{n^2 t_i} \right) \times \log_2 \left( 2 + \sum_j \sigma_j^2 \right)$$

In the equation, $n$ is the number of texts containing the feature word $t_i$ in the text corpus, and $m$ is the total number of feature words $t_k$ appearing in the document. After substituting equation (7) for equation (1) into equation (2), the normalized formula is

$$TFIDF-CV = \frac{tf_{i,j} \times idf_i \times \alpha \times \beta \times \gamma}{\sqrt{\sum_{(i,j)} [f_{i,j} \times idf_i \times \alpha \times \beta \times \gamma]^2}}$$

The above three distribution factors can well reflect the importance of feature words. In an imbalanced classification data set, if the number of documents in which the feature words belong is smaller, the $\alpha$ value is larger. If the feature word mainly exists in a class, its $\beta$ value is larger. If words appear equally in the text, the obtained $\gamma$ value is larger. The larger the values of $\alpha$, $\beta$, and $\gamma$, the greater the weighted end result. From the analysis, it can be concluded that the improved weight calculation formula makes up for the shortcomings of the TFIDF algorithm, adding consideration to the distribution of words in different categories and in a text.

4. Feature Selection and Classification Strategy

4.1. Feature Selection

During the text classification process, word segmentation processing and removal of stop words are performed on the text in the data set. The number of feature words in the obtained feature word set is large, so the dimension of the feature space is large during processing, which will also affect the text classification process. The feature selection process is the dimension reduction process. At present, the keywords selected from the text include information gain IG (Information Gain), mutual information MI (Mutual Information), $\chi^2$ statistic CHI [13] (Chi-square), etc.

In the subsequent experimental verification process, $\chi^2$ statistics were used for feature selection. This method mainly measures the correlation between feature word $t$ and category $C$. It is considered that the relationship between the two closely approximates the $\chi^2$ distribution with a degree of freedom of 1. The larger the $\chi^2$ statistical value, the greater the correlation between feature word $t$ and category $C$. Calculated as follows:
\[ \chi^2_{(t,c)} = \frac{N \times (AE - DB)^2}{(A + D) \times (B + E) \times (A + B) \times (D + E)} \]  

(9)

Among them, \( N \) is the total number of text in the text set, \( A \) represents the number of files in the class \( C \) that own the word \( t \), \( B \) represents the amount of texts that do not belong to class \( C \) but contains the term \( t \), and \( D \) represents the amount of texts that belong to the class \( C \) but does not contain the term \( t \). Number of documents, \( E \) is not in class \( C \) as a document and the word \( t \) does not appear in the document.

This feature selection method is applied to the training set to select feature words in preparation for the comparison between the final improved algorithm TFIDF-CV algorithm and other algorithms.

### 4.2. Classification Strategy—K Nearest Neighbors

K-nearest neighbor classification algorithm is a method to search the hidden information technology from a large number of data [14]. The idea of this method is: In eigenvector space, if most of the k most similar samples of a sample belong to a certain category, the sample also belongs to this category. It inputs instance-based learning, that is, KNN has no obvious course of practice. In this paper, the data set has been processed. The data set has been classified and the key words have been collected. The KNN algorithm system will conduct classification processing directly after receiving new samples. KNN is classified by measuring the distance between different eigenvalues. As for the distance measurement method, Euclidean distance, cosine value, correlation degree, Manhattan distance or others are commonly used. Here Euclidean distance is used, the formula is as follows:

\[ E(x, y) = \sqrt{\sum_{i=0}^{n} (x_i - y_i)^2} \]  

(10)

This formula represents the true distance between two points in n-dimensional space. The Euclidean distance in two and three dimensions is the actual distance between two points.

The original algorithm and the improved algorithm TFIDF-CV are calculated in the training set to obtain key feature words, and the CHI square statistical feature selection method selects features in the test set. After that, the KNN algorithm calculates and performs classification based on the above data in the test set.

### 5. Experimental Results and Analysis

#### 5.1. Experimental Data Set

For the sake of check the effectiveness of the method in this paper, an improved algorithm was implemented in Windows 10 environment using python language. Crawler [15] crawled NetEase news data to build a corpus. The corpus includes finance, sports, automotive, culture and health. The corpus is used as the data set, and the word segmentation processing in the data set preprocessing uses the word segmentation. There are many samples in the data set, and the average number of words in each document is about 1000. At the same time, there is also a data imbalance phenomenon, which is consistent with an imbalanced data set with a large number of words for a single text. The specific number of documents in each category is shown in table 1.

| Type   | Number of training set texts | Number of testing set texts |
|--------|------------------------------|-----------------------------|
| financial | 3000                          | 1500                        |
| sports  | 1860                          | 930                         |
| culture | 2670                          | 1335                        |
| medicine | 1380                          | 690                         |
| cars    | 3800                          | 1900                        |
5.2. Evaluation index
The evaluation index of the classifier model mainly includes the recall rate $R$, precision $P$, and $F$ metrics [16]. Respectively:

$$R = \frac{TP}{TP + FN} \quad (11)$$

$$P = \frac{TP}{TP + FP} \quad (12)$$

$$F = \frac{(\alpha + 1) \times P \times R}{\alpha (P + R)} \quad (13)$$

Among them, $TP$ represents the number that actually belongs to the category and the prediction is correct, $FP$ represents the number that does not actually belong to the category and the prediction is correct, and $FN$ represents the number that actually belongs to the category but the prediction is wrong. There are sometimes contradictions in the recall $R$ and precision $P$ indicators. In this way, the $F$ metric value is weighted and the other two parameters are averaged. When $\alpha = 1$, it is $F_1$. It is clear from the formula that $FN$ balances $P$ and $R$. The value of $FN$ is proportional to the actual effect. In this experiment, $\alpha$ was adjusted to 0.414.

5.3. Analysis of Results
In addition to the original algorithm and the improved algorithm written in the test comparison algorithm, there is an improved TFIDF algorithm under dual parallel computing proposed by Ref. [17]. The final weighted results of different categories are obtained after the training set is executed. According to the weight value, the first 17 feature words are selected as the test control words. At the same time, CHI square statistical feature selection method was used in the test set to select 30 features in each document. Finally, a classification test is performed under the K nearest neighbor classifier. The test results show the accuracy $P$ and $F$ metrics, as shown in table 2.

| Category | TFIDF | Double parallel TFIDF algorithm | TFIDF-CV |
|----------|-------|---------------------------------|----------|
| R/%      | P/%   | F/%    | R/%      | P/%   | F/%    | R/%      | P/%   | F/%    |
| financial | 88    | 92    | 90      | 88    | 96    | 92      | 91    | 97    | 94    |
| sports   | 92    | 80    | 86      | 95    | 85    | 90      | 95    | 87    | 91    |
| culture  | 91    | 83    | 87      | 93    | 87    | 90      | 96    | 90    | 93    |
| medicine | 88    | 88    | 88      | 95    | 87    | 91      | 97    | 89    | 93    |
| cars     | 91    | 93    | 92      | 94    | 98    | 96      | 96    | 98    | 97    |
| mean     | 90.0  | 87.2  | 88.6    | 93    | 90.6  | 91.8    | 95.0  | 92.2  | 93.6  |

The TFIDF, dual-parallel TFIDF algorithm and TFIDF-CV algorithm recall rate $F$ measurement results are compared as follows.

The accuracy is supported by the class distribution factor $\alpha$ and the inter-class distribution factor $\beta$ in the algorithm to distinguish different classes. The higher the accuracy, the less misjudgement of other categories. The recall rate is supported by the variance distribution factor $\gamma$, which is used to find the feature words that really have category characteristics in the class. A high recall rate indicates that feature word selection is more accurate. As can be seen from table 2 and figures 1-3, the accuracy of the improved TFIDF-CV algorithm is significantly improved compared to other categories. At the same time, the $F$ metric value of the feature words extracted by the improved algorithm is obviously superior to the traditional algorithm in all aspects. For the improved TFIDF algorithm under dual parallel computing, the improved algorithm generally improves slightly. The experiments show that the three
distribution factors included in the improved algorithm have different degrees of optimization in their respective scopes.

![Figure 1. Precision.](image)

![Figure 2. F metrics.](image)

![Figure 3. Recall rate.](image)

6. Summary
This paper improves the TFIDF algorithm from the perspectives of category and word frequency, proposes three distribution factors, and then proposes the TFIDF-CV algorithm. The design premise of this algorithm is that the data set belongs to an imbalanced data set with a single text and a large number of words. By calculating, testing, and comparing two algorithms for a large number of text files, it can be seen that the improved method greatly upgrades the precision of the original algorithm's calculation weights.
References

[1] Guo Y 2018 Development status of text classification implementation based on rough set theory Education Forum (46) 241-242.

[2] Su J 2006 Advances in machine learning based text categorization Journal of Software (09) 1848-1859.

[3] Shen H 2006 Comparison and improvements of feature extraction methods for text categorization computer Integrated Manufacturing Systems (03) 222-224.

[4] Chen C-H 2016 Improved TFIDF in big news retrieval: An empirical study Pattern Recognition Letters doi: 10.1016/j.

[5] Zhao S 2017 Improved TFIDF-based question similarity algorithm for the community interlocution systems Transaction of Beijing Institute of Technology 37 (09) 982-985.

[6] Guo H 2013 Research on term weighting algorithm based on information entropy theory Computer Engineering and Applications 49 (10) 140-146.

[7] Cheng L 2020 Research and improvement of TF-IDF algorithm based on information theory Advances in Intelligent Systems and Computing 905.

[8] Chen C-H 2016 Improved TFIDF in big news retrieval: An empirical study Pattern Recognition Letters 11 004.

[9] Xiong Z 2008 Improvement and application to weighting terms based on text classification Computer Engineering and Applications (05) 187-189.

[10] Ramsingh J and Bhuvaneswari V 2018 An efficient map reduce-based hybrid NBC-TFIDF algorithm to mine the public sentiment on diabetes mellitus-A big data approach Journal of King Saud University-Computer and Information Sciences.

[11] Salton G 1998 Term-weighting approaches in automatic text retrieval Information Processing and Management 24 (5) 513-523.

[12] Yang X 2013 Discussion of variance analysis: The single factor variance analysis Experimental Science and Technology 11 (01) 41-43.

[13] Song Z 2018 Multi-level feature selection mechanism based on MapReduce Computer Science 45 (S2) 468-473+479.

[14] Wang Y 2018 KNN similarity graph algorithm based on heap and neighborhood coexistence Computer Science 45 (05) 196-200+227.

[15] Zhou Z 2014 Data crawler for Sina WeiBo based on Python Journal of Computer Applications 34 (11) 3131-3134.

[16] Zhou Z 2016 Machine Learning (Tsinghua University Press).

[17] Sun Y 2016 TFIDF algorithm based on dual parallel calculation mod Computer Engineering and Design 37 (11) 3016-3021.