An Improved Feature Weighting Method for Text Classification

Qiaoyan Kuang\(^1\), Xiaoming Xu\(^2\)

\(^1\) College of Computer, Hunan International Economics University, 410205 Changsha, China
\(^2\) College of Continuing Education, Hunan International Economics University, 410205 Changsha, China.
{qiaoyankuang@126.com, ceoxoxo@163.com}
doi : 10.4156/aiss.vol3.issue7.40

Abstract

Feature extraction is the important prerequisite of classifying text effectively and automatically. TF•IDF is widely used to express the text feature weight. But it has some problems. TF•IDF can't reflect the distribution of terms in the text, and then can't reflect the importance degree and the difference between categories. This paper proposes a new feature weighting method—TF•IDF•Ci, to which a new weight Ci is added to express the differences between classes on the base of original TF•IDF. After combining TF•IDF•Ci and specific classification algorithm, it always get a larger macro F1 value than of TF•IDF. Meanwhile, the standard deviation of the classification index of the TF•IDF•Ci is much smaller than that of TF•IDF. That shows TF•IDF•Ci not only improve the classification precision but also decreases the sensitivity towards feature dimensions to some extent.

Keywords: Text Classification, TF•IDF; TF•IDF•Ci, Feature Extraction, Feature Weighting, Feature Selection

1. Introduction

With the fast popularization of internet and rapid development of digital media technology, thousands of new texts emerge on the internet. To manage such a huge amount of texts, classification needs urgent settlement. So study on automatic classification of texts through computer has become a project with practical significance to the data mining and artificial intelligence area.

The aim of text automatic classification is to handle the unknown texts and to determine which predefined categories they belong to [1]. Feature extraction is the premise and base to realize the effective automatic classification of texts [2-4]. If the extracted feature items can't express the content of the text and reflect the differences between categories properly, then automatic classification makes no sense. Represented by feature item through word segmentation algorithm and word frequency statistics directly, the dimension of the vector shall be very large [5]. So we can't classify using all the words in the text. If we adopt the untreated text vectors to classify the text, it will not only cost much time, to reduce the efficiency of the whole process, but also affects the precision of classification algorithm. The results as above are not always so satisfied. So we have to purify the text vectors on the base of maintaining its original meaning to find the text feature which can represent its text feature category best. To solve this problem, the most effective way is to reduce dimension through feature extraction [6-8] [12]. The common way is to pick out the terms which make great contributions to text classification according to some selection strategy. The conventional selection strategy includes TF•IDF, DF, IG, MI, CHI, ECE and so on [9-13].

This paper proposes a new feature weighting method TF•IDF•Ci on the base of the original TF•IDF. The extracted feature can represent the content of the text better and has a better distinguished ability. Experiments show that the proposed method can pick out the terms which make great contributions to text classification and improve the classification precision.

2. TF•IDF Method

TF•IDF is the one of the most effective ways to calculate term weight. TF•IDF equals TF multiplies by IDF, in which TF is short for term frequency that is used to calculate the describing ability of the
term; and IDF is short for inverse document frequency which is used to calculate the distinguishing ability of the term [8].

\[
IDF = \log \frac{N}{n}
\] (1)

Where N is the text total of all categories, n is the number of the texts which include term t. The main idea of IDF is that the smaller n is, the larger the IDF is, and term t will have a better category distinguishing ability.

The main idea of TF•IDF can be expressed as follows: if a word or phrase has a high TF value and hardly appears in other texts, we can say it has a good category distinguishing ability and fits for classification.

TF•IDF is based on the assumption that the word which can distinguish the texts should be the one that appears frequently in the text but reversely in other texts. So if we take TF as measure, the feature space coordinate system can reflect the character of analogical texts. Besides, taking distinguish ability of the term on different texts into consideration, we can say that the larger TF is, the better distinguishing ability on different texts it has according to TF•IDF. So IDF is introduced in this paper. We take the product of TF and IDF as the measure of feature space coordinate system, through which the weight TF is adjusted. The purpose of adjusting weight is to stress the key term and suppress the subordinate term. Finally the weights of all terms are sorted. Two methods can be used to select the terms: (1) choosing the n terms, which have the biggest weights; (2) choosing these terms, the weights of which are bigger than a certain threshold. Some experiments show that 4-7 key words fit for manual selection and 10-15 key words have the best coverage and specificity for computer selection.

3. Improvement of TF•IDF Method

3.1 Problem of TF•IDF Method

TF•IDF is generally accepted as an effective way for feature extraction. It introduces IDF weight to improve the feature distinguishing ability of terms. That is to say, TF•IDF is based on the assumption that the words which are most meaningful to distinguish text should be the one that appears in the text frequently but reversely in other texts. That is to say, the terms which can represent the feature of the text should meet both of the following requirements:

1 ) the term has a high frequency in the text, i.e., a large TF value;
2 ) the term has a low frequency in other texts, i.e., a small IDF value.

The above requirements haven’t considered the distribution of feature terms between and within categories. It is embodied in the following two aspects:

First TF : If we simply use the terms with high TF to represent the feature of the text, the terms with low TF which can also represent the feature of the text very well are likely to be neglected. In fact if the TF of term in this text is low but high in certain category(not all the texts), the terms can represent the feature of the text very well.

Then IDF : According to TF•IDF, the terms which have low frequency in other texts of the whole text collection can represent the feature of the text. But the range of the “other texts” defined by this algorithm is too wide. In fact if the TF of terms in the text is low in other texts of the whole text collection but high in the texts of a certain category, the terms can also represent the feature of the text very well.

The two aspects lay in that the range of TF defined by TF•IDF is too wide and neglects the frequency in the texts of a certain category. That is to say, the structure of TF multiplies by IDF can’t
reflect the distribution of terms in a certain category and then can’t reflect the importance degree and difference between categories.

3.2 Improvement of TF-IDF Method

The original TF-IDF is TF multiplies by IDF, where TF and IDF is short for term frequency and inverse document frequency respectively. Because of the problems of TF-IDF as described above, we have to add a weight to the original TF-IDF. The added weight considers the frequency of the term, which is in a particular category in the whole text collection, rather than simply consider the frequency of the term which is in the other documents of the whole text collection.

Suppose N is the total number of texts, n is the number of the texts containing term t, m is the maximum number of texts containing term t in a certain category.

Therefore, to improve the distinguishing ability between categories, i.e. the category to which the text containing term t belongs and other categories, we introduce a new weight \( C_i = \frac{1}{n-m+1} \).

Then the formula is modified as:

\[
TF \cdot IDF \cdot C_i = TF \times IDF \times C_i
\]

Where \( (n-m) \) is the Difference-value (D-value) between the number of all texts containing term t and the maximum number of texts containing term t in a certain category. When the number of the texts in a category containing term t is large, the number of the texts in the other categories containing term t, i.e. \( (n-m) \), is small. Then the term can represent the feature of the category of the texts containing the largest number of term t, so the weighting value is large, i.e. the feature representation ability of term t is inversely proportional to the number of the texts in all categories except the category containing the largest number of term t. Because term t is exclusively included by a single category, i.e. m equals n, the denominator should be \((n-m+1)\).

4. Experiments and Result

In our experiments we mainly use KNN (K Nearest Neighbor) as text classifier. KNN is a conventional pattern recognition method and widely used in text classification. It is good at precision and recall rate.

KNN searches for K samples most analogical to the samples to be classified among the known samples and predicts the category of the unknown samples on the base of the K known samples. An easy predicting way is to assume the category of the unknown samples as the category containing maximum examples among the K samples. When the classifier is set, we only need compare the similarity of the measured text and category classifier to confirm whether the measured text belongs to the category.

To measure the similarity of two texts, we always measure similarity of their vectors. The similarity of vectors can be measured by the cosine from the angle of their vectors. The larger the cosine is, the higher similarity is. When the vectors are \( d_i(a'_1, a'_2, \cdots, a'_n) \) and \( d_j(a'_1, a'_2, \cdots, a'_n) \), the cosine of their angle is:

\[
\cos(d_i,d_j) = \frac{\sum_{i=1}^{n} (a'_i \cdot a'_i)}{\sqrt{\sum_{i=1}^{n} (a'_i)^2} \times \sqrt{\sum_{i=1}^{n} (a'_j)^2}}
\]
An Improved Feature Weighting Method for Text Classification
Qiaoyan Kuang, Xiaoming Xu
Advances in Information Sciences and Service Sciences, Volume3, Number7, August 2011

If \( \cos(d_i, d_j) > \bar{\sigma} \), we can say \( d_i \) is similar to \( d_j \), and vice versa. Here \( \bar{\sigma} \) is the similarity threshold value, also an empirical value.

In our experiments, we adopted Chinese corpus (download from international database Center, department of computer information and technology, Fudan University, China), which includes 19637 texts covering human resources, health, entertainment, mortgage, education, automobile, computer, science, finance and economics and etc. We chose five categories of them. Both training and testing sets include 5000 texts. That is to say each category includes 1000 training texts and 1000 testing texts.

To evaluate the classification performance thoroughly and the precision, we use macro F1 as the evaluation index:

\[
Macro\ _F1 = \sum_{i=1}^{m} \frac{N_i}{N} \times F1 = \sum_{i=1}^{m} \frac{N_i}{N} \times \frac{2 \times precision_i \times recall_i}{precision_i + recall_i}
\]

Here, \( N_i \) is the number of texts of category \( i \), \( N \) is the total number of measured texts. \( \text{Precision}_i \) and \( \text{recall}_i \) are the precision and recall rate of category \( i \), and have \( m \) categories.

4.1. Comparison of the macro F1 value of two weighting methods

When choosing the features of the texts, we firstly choose features number of 500 to 20000 through the weight of evidence for text, then take feature weight by \( \text{TF} \cdot \text{IDF} \cdot \text{Ci} \) respectively. The classification performance of KNN classifier and Genetic classifier is listed in Table 1:

<table>
<thead>
<tr>
<th>Number</th>
<th>( \text{maF1 (TF-IDF-Ci AND KNN)} )</th>
<th>( \text{maF1 (TF-IDF-Ci AND Genetic)} )</th>
<th>( \text{maF1 (TF-IDF AND KNN)} )</th>
<th>( \text{maF1 (TF-IDF AND Genetic)} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>92.254</td>
<td>92.82</td>
<td>87.315</td>
<td>88.451</td>
</tr>
<tr>
<td>1000</td>
<td>91.763</td>
<td>91.718</td>
<td>88.374</td>
<td>88.945</td>
</tr>
<tr>
<td>2000</td>
<td>92.854</td>
<td>92.975</td>
<td>89.012</td>
<td>90.136</td>
</tr>
<tr>
<td>4000</td>
<td>92.350</td>
<td>92.869</td>
<td>90.457</td>
<td>89.854</td>
</tr>
<tr>
<td>6000</td>
<td>91.892</td>
<td>91.452</td>
<td>88.425</td>
<td>87.596</td>
</tr>
<tr>
<td>8000</td>
<td>92.806</td>
<td>92.363</td>
<td>89.689</td>
<td>89.355</td>
</tr>
<tr>
<td>10000</td>
<td>91.935</td>
<td>92.444</td>
<td>88.032</td>
<td>87.980</td>
</tr>
<tr>
<td>12000</td>
<td>89.982</td>
<td>91.763</td>
<td>86.452</td>
<td>86.265</td>
</tr>
<tr>
<td>15000</td>
<td>91.961</td>
<td>90.754</td>
<td>87.325</td>
<td>85.547</td>
</tr>
<tr>
<td>18000</td>
<td>91.742</td>
<td>90.553</td>
<td>84.570</td>
<td>84.663</td>
</tr>
<tr>
<td>20000</td>
<td>91.818</td>
<td>91.318</td>
<td>87.016</td>
<td>87.308</td>
</tr>
<tr>
<td>x(mean)</td>
<td>91.942</td>
<td>91.912</td>
<td>87.879</td>
<td>87.827</td>
</tr>
<tr>
<td>S(standard deviation)</td>
<td>0.760</td>
<td>0.847</td>
<td>1.616</td>
<td>1.690</td>
</tr>
</tbody>
</table>

Of Table 1, we can find that:

(1) With the different numbers of features, the macro F1 of TF-IDF-Ci with KNN classifier is larger than that of it with Genetic classifier.

(2) With the different numbers of features, the macro F1 of TF-IDF with KNN classifier is larger than that of it with Genetic classifier.
With the different numbers of features, the macro F1 of TF-IDF-Ci is larger than that of TF-IDF.

(4) when TF-IDF choose 4000 dimension features, its macro F1 value is 90.457, which is the best value, but still 2.103 percentage lower than that of TF-IDF-Ci.

(5) After calculating the mean value and standard deviation of macro F1, we find that the standard deviation of TF-IDF is much larger than that of TF-IDF-Ci. This proves that the classified precision of TF-IDF-Ci keeps relatively stable with the change of dimensions, which reduces its sensitivity to feature dimension to some extent.

The result proves that our modification is effective. TF-IDF-Ci not only improves the classification precision, but also reduces its sensitivity to feature dimension to some extent. It is especially useful to the classifiers sensitive to feature dimensions.

4.2. Comparison of the different feature selection methods

When choosing the features of the texts, we firstly choose features number of 500 to 20000 through the MI, IG, \( \chi^2 \) and Cross Entropy, then take feature weight by TF-IDF and TF-IDF-Ci respectively. The classification performance of KNN classifier is listed in Table 1:

<table>
<thead>
<tr>
<th>Methods</th>
<th>( maF1 ) (TFIDF)</th>
<th>( maF1 ) (TF-IDF ( \times ) CI)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mutlnfo</td>
<td>x</td>
<td>78.29</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>2.134</td>
</tr>
<tr>
<td>InfGain</td>
<td>x</td>
<td>89.12</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>2.011</td>
</tr>
<tr>
<td>( \chi^2 )</td>
<td>x</td>
<td>88.56</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>1.958</td>
</tr>
<tr>
<td>CroE nTxt</td>
<td>x</td>
<td>89.61</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>1.982</td>
</tr>
</tbody>
</table>

Of Table 1, we can find that the macro F1 of TF-IDF-Ci is larger than that of TF-IDF, no matter what feature selection method were selected. After calculating the mean value and standard deviation of macro F1, we find that the standard deviation of TF-IDF is much larger than that of TF-IDF-Ci. This proves that the classified precision of TF-IDF-Ci keeps relatively stable with the change of dimensions, which reduces its sensitivity to feature dimension to some extent. The result proves that our modification is effective. TF-IDF-Ci not only improves the classification precision, but also reduces its sensitivity to feature dimension to some extent. It is especially useful to the classifiers sensitive to feature dimensions.
4.3. Comparison of the similar methods

Compared with other similar methods, the results are shown in Table 3.

<table>
<thead>
<tr>
<th>Method</th>
<th>maF1</th>
<th>x</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>The proposed method</td>
<td>91.942</td>
<td>0.760</td>
<td></td>
</tr>
<tr>
<td>Zhen Wang’s method [8]</td>
<td>74.93</td>
<td>1.583</td>
<td></td>
</tr>
<tr>
<td>Yongmin Lin’s method [9]</td>
<td>91.911</td>
<td>0.854</td>
<td></td>
</tr>
<tr>
<td>The proposed method</td>
<td>91.912</td>
<td>0.847</td>
<td></td>
</tr>
<tr>
<td>Yufang Zhang’s method [10]</td>
<td>78.64</td>
<td>1.452</td>
<td></td>
</tr>
</tbody>
</table>

Of Table 3, we can find that the macro F1 of the proposed method, which combined with KNN classifier and Genetic classifier, is larger than that of other methods, no matter what feature selection methods were selected. The standard deviation of the proposed method, which combined with KNN classifier and Genetic classifier, is much smaller than that of other methods. The result proves that our modification is effective. TF-IDF·Ci not only improves the classification precision, but also reduces its sensitivity to feature dimension to some extent.

5. Conclusions and Future work

This paper proposes a new feature weighting method -TF-IDF·Ci on the base of original TF-IDF. It improves the distinguishing ability between categories through adding a new weight Ci.

We have done experiment by combining the method in this paper and classification algorithm. After comparing the results with the other method, we can find macro F1 of TF-IDF·Ci is always larger than that of TF-IDF. At the same time, the standard deviation of the classification index of the TF·IDF·Ci is much smaller than that of TF-IDF. That shows that TF-IDF·Ci not only improves the classification precision but also decreases the sensitivity towards feature dimension to some extent, which is specially useful to the classifiers sensitive to feature dimension.

The next work is to take the fingerprint of the text through TF-IDF·Ci for the copy detection of the text.

6. Acknowledgements

This paper was supported by Science and Technology Bureau Foundation of Changsha City (No. K10ZD030-13).

7. References

[6] Jiawei Han, Jian Pei, and Yiwen Yin, “Mining Frequent Patterns without Candidate Generation”, Proc.2000 ACM-SIGMOD Int.Conf. Management of Data (SIGMOD'00), pp. 1-12, 2000.