2012 International Conference on Computer Technology and Science (ICCTS 2012)
IPCSIT vol. 47 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V47.9

An improvement of TFIDF weighting in text categorization

Mingyong Liu[1+] and Jiangang Yang [2]

[1] Zhejiang University

Abstract. A main problem in text categorization is how to improve the classification accuracy. In this paper, in order to improve the accuracy, we propose a new weighting method named TF-IDF-CF based on TF-IDF which is a widely used weighting method in text categorization. From the experiment results, we can see this method can achieve very good results.

Keywords: text categorization, feature selection, CHI square statistics, TFIDF, categorization accuracy.

1. Introduction

With the rapid growth of online information, how to process tons of text efficiently becomes a hot research topic, text categorization is one of the key tasks among them. Text categorization is to assign new documents to pre-existing category, and it has been widely used in many areas like information retrieve, email classification, junk email filtering, topic spotting.

In recent years, most research has been focused on finding new categorization algorithms, little research has been done on improvement of document representation models, which comes from information retrieval. There're 3 traditional models: vector space model[1], probabilistic model, inference network model, vector space model is the most widely used model among them.

In vector space model, feature is represented as weighting using numbers, there're some common used weighting methods, such as Boolean weighting, frequency weighting, TF-IDF weighting, TFC weighting[2], LTC[9] weighting, entropy weighting, TF-IDF weighting is the most widely used one among them.

In this paper, we propose an improvement of TF-IDF weighting on vector space model, TF-IDF considers both the term frequency and inverse document frequency, in this method, if the term frequency is high and the term only appears in a little part of documents, then this term has a very good differentiate ability, this approach emphasizes the ability to differentiate different classes more, whereas it ignores the fact that the term that frequently appears in the documents belonging to the same class, can represent the characteristic of that class more. So we introduce a new parameter to represent the in-class characteristic, and then we conducted some experiments to compare the effects, the result tells us this improvement has better accuracy.

2. Text Categorization Steps

Generally, text categorization often includes 5 main steps: document preprocessing, document representation, dimension reduction, model training, testing and evaluation[3].

2.1. Document Preprocessing

In this step, we remove html tags, rare words, stopping words, and may need to do some stemming, this is simple in English, but difficult in chinese, japanese and some other languages.

[1] Corresponding author. Tel.: +86 18768142091.
E-mail address: liumy601@163.com.

2.2. Document Representation

Before doing classification, we need to transform the documents into a format that computer can recognize, vector space model(VSM) is most commonly used method. This model takes the document as a multi-dimension vector, and the feature selected from the dataset as a dimension of this vector.

2.3. Dimension Reduction

Because in documents, there're tens of thousands of words, if we choose all of them as features, then it'll be infeasible to do the classification, as the computer can't process such amount of data. So we need to select those most meaningful and representative features for classification, the most commonly used selection methods contains CHI square statistics[4], information gain, mutual information, document frequency, latent semantic analysis.

2.4. Model Training

This is the most important part of text categorization. It includes choosing some documents from corpus to comprise the training set, performs the learning on the training set, and then generates the model.

2.5. Testing and Evaluation

This step uses the model generated from step 4, and performs the classification on the testing set, then chooses appropriate index to do evaluations.

3. TF-IDF

In vector space model, TF-IDF is a widely used weighting method, which was firstly introduced from information retrieval. TF-IDF[5,6](Term Frequency-inverse Document Frequency), puts weighting to a term based on its inverse document frequency. It means that if the more documents a term appears, the less important that term will be, and the weighting will be less. It can be depicted as this:

$$a_{ij} = tf_{ij} * \log(\frac{N}{n_j}) \tag{1}$$

In formula (1), tf_{ij} represents the term frequency of term j in document i, N represents the total number of documents in the dataset, n_j represents the number of documents that term i appears.

When N equals n_j, then a_{ij} becomes zero, this often appears in small dataset, so we need to apply some smoothing[7] techniques to improve formula (1) as following:

$$a_{ij} = \log(tf_{ij} + 1.0) * \log(\frac{N + 1.0}{n_j}) \tag{2}$$

4. TF-IDF-CF

Regarding the shortcomings TF-IDF has, we introduce a new parameter to represent the in-class characteristics, and we call this class frequency, which calculates the term frequency in documents within one class. Then we rename this new weighting method to **TF-IDF-CF**, its formula is based on (2):

$$a_{ij} = \log(tf_{ij+1.0}) * \log(\frac{N + 1.0}{n_j}) * \frac{n_{cij}}{N_{ci}} \tag{3}$$

n_{cij} represents the number of documents where term j appears within the same class c document i belongs to, N_{ci} represents the number of documents within the same class c document i belongs to.

5. Experiment and Analysis

In the experiment, we choose to use the commonly used datasets Reuters-21578 and 20newsgroup. Before proceeding, we conduct some preprocessing like removing html tags, filtering invalid characters, removing stopping words, and then lower all words. After this processing, for Reuters-21578, we choose 6088 training samples, 2800 testing samples, and 59 classes. For 20newsgroup, we choose 8000 training samples, 2000 testing samples, and 20 classes. Then we use CHI square statistics feature selection method to

select 1000 features, and then we conduct the experiments using TF-IDF, TF-IDF-CF, LTC, TFC weighing method separately on some commonly used classifiers[8] Naïve Bayes, Bayes Network, KNN, SVM. After the experiment, we compare result of TF-IDF-CF with TF-IDF, LTC, TFC.

5.1. CHI square statistics

CHI square statistics[4] is a very useful feature selection method in text categorization, it can measure the correlation between feature and class. Let A be the times both feature t and class c exists, B be the times feature t exists, but class c doesn't exist, C be the times feature t doesn't exist, but class c exists, D be the times both feature t and class c doesn't exist, N be the total number of the training samples. Then CHI square statistics can be depicted as:

$$\chi^2(t,c) = \frac{N*(AD-BC)^2}{(A+C)*(B+D)*(A+B)*(C+D)} \tag{4}$$

5.2. TFC and LTC

TF-IDF doesn't consider the effect of the length of different documents on weighting, in order to include such effect, **TFC**[2] is proposed and it's actually the normalization of formula (1).

$$a_{ij} = \frac{tf_{ij} * \log(\frac{N}{n_j})}{\sqrt{\sum_{p=1}^{M} tf_{ip} * \log(\frac{N}{n_j})}} \tag{5}$$

LTC[9] is a different format of TF-IDF, it considers the limit of small datasets, and it's actually the normalization of formula (2).

$$a_{ij} = \frac{\log(tf_{ij}+1.0) * \log(\frac{N}{n_j})}{\sqrt{\sum_{p=1}^{M} [\log(tf_{ip}+1.0) * \log(\frac{N}{n_p})]^2}} \tag{6}$$

5.3. Experiment

Based on the two datasets, we use CHI square statistics method to select 1000 features, then we conduct the experiments **on** a well known data mining tool named WEKA using some common used algorithms like Naïve Bayes, Bayes Network, KNN, SVM, we only consider classification accuracy when comparing the result:

Table 1 Results

Weighting Method	Naïve Bayes		Bayes Network		KNN		SVM	
	Reuters	20news	Reuters	20news	Reuters	20news	Reuters	20news
TFC	67.1%	62.3%	72.2%	67.8%	70.7%	58.9%	81.3%	63.8%
LTC	62.9%	61.8%	74.7%	63.4%	71.2%	53.1%	83.1%	67.2%
TF-IDF	61.6%	61.9%	76.9%	65.3%	72.8%	55.3%	84.7%	69.1%
TF-IDF-CF	88.6%	77.1%	91.4%	77.7%	81.4%	64.9%	92.8%	78.7%

5.4. Analysis

From experiment results on Table 1, we can see our improved TF-IDF-CF weighting method has the best precision in both Reuters-21578 and 20newsgroup, and the precision has greatly increased compared with

original TF-IDF weighting method. Although TFC and LTC have better results than TF-IDF on some classifiers like Naïve Bayes, it's not very meaningful like TF-IDF, so they're not usually used to calculate weighting. The reason why our new method increases the precision greatly is TF-IDF only emphasizes the ability to differentiate the different classes, but undervalues the ability to represent the class itself. The more one term appears in the documents of one class, the more important that term will be to represent that class. From the theory and experiment, we can see this improvement can achieve a better accuracy.

6. Conclusion

Text categorization is a hot research topic in current information retrieval, and is an important branch of data mining and information retrieval. How to improve the classification accuracy is an important topic in text categorization, in order to solve this problem, much research has been done to find new classifiers which will improve the accuracy, whereas this paper tries to improve the accuracy by proposing an improvement on TF-IDF weighting method. From the experiments, we can see this improvement increases the accuracy significantly, therefore we think this improvement is promising.

7. References

[1] Ami Singhal. Modern Information Retrieval: A Brief Overview. IEEE, 2001: 2-4.

[2] GERARD SALTON, CHRISTOPHER BUCKLEY. TERM-WEIGHTING APPROACHES IN AUTOMATIC TEXT RETRIEVAL. Information processing and management, 1988: 3-6.

[3] FABRIZIO SEBASTIANI. Text categorization[M]//Alessandro Zanasi. Text mining and its applications. WIT Press, Southampton, UK, 2005: 110-120.

[4] YANG Y, PEDERSEN J Q. A comparative study on feature selection in text categorization. Proceedings of the 14th International Conference on Machine Learning (ICML), 1997: 2-3.

[5] SALTON G, WONG A, YANG C S. A vector space model for automated indexing. Communications of the ACM, 1975: 1-8.

[6] SALTONG, MCGILLC. An introduction to modern information retrieval. McGraw Hill, 1983.

[7] Stanley F Chen, Joshua Goodman. An Empirical Study of Smoothing Techniques for Language modeling. Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics, 1996: 3-9.

[8] FABRIZIO SEBASTIANI. Machine Learning in Automated Text Categorization. ACM Computing Surveys, Vol. 34, No. 1, March 2002: 18-33.

[9] NIGAM K, LAFFERTY J, MCCALLUM A. Using maximum entropy for text classification. Proceedings of the IJCAI-99 Workshop on Information Filtering, Stockholm, Sweden, 1999: 58-65.