Optimal Document Representation Strategy for Supervised Term Weighting Schemes in Automatic Text Categorization

Longjia Jia[1, a], and Bangzuo Zhang[2, b]

[1]School of Mathematics and Statistics, Northeast Normal University, Changchun 130024, China

[2]School of Computer Science and Information Technology, Northeast Normal University, Changchun 130024, China

[a]jialongjia@nenu.edu.cn; [b]zhangbz@nenu.edu.cn

Keywords: Optimal document representation; Term weighting; Text Categorization; Machine learning

Abstract: Term weighting is a strategy that assigns weights to terms in order to improve the performance of text categorization. In this paper, we propose a document representation strategy for supervised text classification named the optimal document representation strategy for supervised term weighting schemes (*ODRS*), which can get the optimal term weighting vector in many different vectors. The main idea of *ODRS* is that by proposing optimal function and introducing the importance of categories and terms on training set to find the optimal parameters and then this optimal model will be applied to test set. In the experiments, we investigate the effects of *ODRS* on the 20 Newsgroups and Reuters21578 datasets using the *SVM* as classifier. The results show that the *ODRS* outperforms other text representation strategy schemes, such as Document Max, Document Two Max and global policy.

1. Introduction

Text categorization (TC) is the task of automatically classifying unlabeled natural language documents into a predefined set of semantic categories. As the first and a vital step, text representation converts the content of a textual document into a compact format so that the document can be recognized and classified by classifiers [1]. The vector space model (VSM) is the most widely used text representation model in text categorization. In VSM, a document is represented as a vector in term spaces, such as $d=\{t_1, t_2, \ldots, t_n\}$, where n is the total number of features. The value of t_i between [0,1] represents how much the term t_i contributes to the semantics of document d. The terms in VSM are extracted from training set. They can be words, phrases, or n-grams, etc.

Each document in datasets is represented as a corresponding vector in vector space. The elements in each vector are weighted by term weighting methods. Most study of term weighting methods for TC has showed that supervised term weighting methods are superior to unsupervised term weighting methods [2]. The difference is that supervised term weighting methods use class information in training set. However, most of the existing methods did not discuss the representation of test documents for supervised term weighting methods [3].

There are two major strategies, local policy and global policy. In local policy, each test document in the independent binary classification task will be represented as a single vector. This means that the vector representation of each document is not an independent vector but a corresponding vector collection which combines with specific binary classification task. Global policy has been widely used. Each document will have a global independent representation. In most classification tasks, each document is generally assigned to one category and labeled with the most similar class label. Thus, most of the classification tasks are regarded as single label task and use global policy. Global policy is defined as Eq. 1.

  DOI: 10.25236/iciss.2019.026

$$TW(t) = \max_{i=1}^{|C|} TW(t, c_i) \tag{1}$$

In Eq. 1, *TW(t)* is the final weight of a term *t*; *TW(t,c_i)* is weight of term *t* in category c_i obtained with supervised term weighting methods. |*C*| is the number of categories. In the process of initial representation, a test document can be represented as |*C*| different vectors. After using appropriate selection policy, it can be represented as one vector which well describes the document. Global policy selects the maximum term value among all categories for each term. Although this method is effective in some cases, but not sure if it has the ability to select the most effective term weighting vector for current test samples [4,5].

Due to above aspects, is there an optimal document representation strategy for supervised term weighting schemes, and, if yes, which one is expected to achieve the best performance, and, if no, can we propose a strategy for this work? This is the question we wish to address in this study. To the best of our knowledge, we have not found any research work on this issue.

In this paper, we investigate several well-known document representation strategy, including "global policy", *W-Max*, *D-Max*, and *D-TMax* for supervised term weighting schemes. Since we have not discovered any similar work presently, this investigation is significant and valuable in document representation strategy for supervised term weighting schemes in automatic text categorization. At the same time, an optimal document representation strategy for supervised term weighting schemes is proposed. The document representation strategies are tested on two famous document collections, i.e., Reuters-21578 and 20 Newsgroups.

The remainder of this paper is organized as follows. We briefly review several document representation strategies in Section 2. Section 3 introduces our optimal document representation strategy for supervised term weighting schemes. We show experimental results in Section 4, and finally, we draw conclusions in Section 5.

2. A Brief Review of Document Representation Strategies

In the scenario of text categorization, an indexing procedure which converts the raw document into a vector representation is usually necessary since text documents cannot be directly interpreted by a classifier. There are some strategies for the representation of a document. Document representation is thereby one of the essential components for the construction of a classifier. In this section, we briefly review several document representation strategies.

Besides local policy and global policy, Younghoong Ko proposed the following three solutions for this problem, i.e., *W-Max*, *D-Max* and *D-TMax* [3]. They are described as follows.

1) *W-Max*: each term's value of term weighting vector will be replaced by the maximum value of the corresponding dimension's term weight in all categories. After comparing with global policy, we may find that they have the same idea.

2) *D-Max*: the sum of all term weights in each term weighting vector is first calculated and then one term weighting vector with the maximum sum value is selected as the document representation vector.

3) *D-TMax*: the sum of all term weights in each term weighting vector is calculated and then two term weighting vectors with the two largest sum values are selected. Then the term weighting vector is constructed by choosing the higher term weighting value from the selected two term weighting vectors for each corresponding dimension's term weight.

3. Methodology

How can we know the effect of document representation method before we choose it? In other words, when selecting a document representation strategy for an unknown data set, which method should we choose? In this study, we will propose a method, which can select appropriate method for the unknown data set. No matter the data set is uniform or not, it will use traversal method to find the optimal strategy on training set.

In Table 1, we first present the notation used in the theories of term weighting.

Table 1 Notations used to formulate term weighting schemes

Notation	*Description*
a	Number of training documents in the positive category containing term t_i.
b	Number of training documents in the positive category which do not contain term t_i.
c	Number of training documents in the negative category containing term t_i.
d	Number of training documents in the negative category which do not contain term t_i.

The improper selecting of document representation strategy would lead to the problem of inappropriate to assign the weight to terms. A test document can be first represented as $|C|$ different vectors by using estimated distribution of each category. For some categories, the weight they assign to terms would has a negative impact on the role of terms in classification. To illustrate this, suppose the training set is skewed with 19 documents, 5 terms and 5 categories. The relationship between term, document, and category is shown in the Table 2. The number in Table 2 represents the times that a term occurs in a document.

Table 2 The relationship between term, document, and category

category	*document*	t_1	t_2	t_3	t_4	t_5
C_1	d_1	0	0	2	19	3
C_1	d_2	0	1	3	0	3
C_1	d_3	0	0	5	16	2
C_1	d_4	4	0	1	15	2
C_1	d_5	0	0	2	18	3
C_2	d_6	0	0	2	14	3
C_2	d_7	0	1	3	0	3
C_2	d_8	0	0	5	13	2
C_2	d_9	4	0	1	11	2
C_2	d_{10}	0	0	2	17	3
C_3	d_{11}	0	0	3	0	3
C_3	d_{12}	0	0	1	1	3
C_3	d_{13}	0	1	1	0	2
C_4	d_{14}	1	99	3	2	1
C_4	d_{15}	2	99	1	1	1
C_4	d_{16}	1	99	1	2	1
C_5	d_{17}	4	0	3	0	3
C_5	d_{18}	4	0	1	0	3
C_5	d_{19}	4	1	1	0	2

According to some existing term schemes such as $tf*rf = tf*log\ (2+a/max(1,c))$, the term weights of t_1 to t_5 for each category are shown in the Table 3.

Table 3 The term weights of t1 to t5 for each category

category	t_1	t_2	t_3	t_4	t_5
C_1	1.1155	1.1699	1.2538	1.3626	1.2538
C_2	1.1155	1.1699	1.2538	1.3626	1.2538
C_3	1.0000	1.1699	1.1375	1.0704	1.1375
C_4	1.2630	1.4510	1.0875	1.1520	1.0875
C_5	1.4594	1.0000	1.1375	1.0000	1.1375

In Younghoong Ko's methods, *D-TMax* selects the two largest sum values. For multiclass classification problems, whether can choose more categories to get good performance? Now we will select 1 to 5 categories to test this hypothesis. When choosing 1 or 2 categories, it is called "*D-Max*" (Document Max) or "*D-TMax*" (Document Two Max) [4]. Based on this rule, we named 3, 4 and 5 categories as "*D-3Max*" (Document Three Max), "*D-4Max*" (Document Four Max) and "*D-5Max*" (Document Five Max). For multiclass text categorization, we named it "*D-NMax*" (Document Number Max) in this study. The number of selected categories and corresponding results are shown in the Table 4.

Table 4 The number of selected categories and corresponding results

method	t_1	t_2	t_3	t_4	t_5
D-Max	1.1155	1.1699	1.2538	1.3626	1.2538
D-TMax	1.1155	1.1699	1.2538	1.3626	1.2538
D-3Max	1.2630	1.4510	1.2538	1.3626	1.2538
D-4Max	**1.4594**	1.4510	1.2538	1.3626	1.2538
D-5Max	**1.4594**	1.4510	1.2538	1.3626	1.2538

The result in bold in the Table 4 violates our intuition that the weight of t_1, t_2 and t_4 should be large, and the weight of t_1 should be relatively small. Since t_1 appear with a low frequency in documents compared to t_2 and t_4. Another unreasonable observation is that t_1 in some documents (d_4 and d_9) in category 1 and category 2 also have same frequency when compared to t_1 in the documents in category 5. After observation, we can find that the results of *D-Max*, *D-TMax* and *D-5Max* cannot boost the performance of text categorization. The results from *D-3Max* are consistent with our intuition that the weight of t_1, t_2 and t_4 should be large, and the weight of t_1 should be relatively small. In order to overcome the shortcomings of Younghoong Ko's methods, in this section we explain our proposed *ODRS* method, which will choose the right method to appropriately weigh the contribution of each term. The *ODRS* can select the appropriate "*N*" (in *D-NMax*) to enhance the performance of text categorization.

Table 5 optimal document representation strategy for supervised term weighting schemes

Algorithm 1: optimal document representation strategy for supervised term weighting schemes

Input:
fea: feature matrix of training set
gnd: a vector of labels for documents in training set
Output:
selectedC: the most appropriate *N* value (in *D-NMax*) for current dataset
Local variables
$|C|$: total number of categories;
M: total number of features;
termWeightingVec1: the set of $|C|$ original term weighting vectors;
$termWeightingVec1_i$: *i*-th vector of the original term weighting vectors;
$termWeightingVec2_i$: *i*-th vector of the reconstructed term weighting vectors;
$sumVec_i$: sum value of all terms in *i*-th term weighting vector;
sortSum: sorted list of each sum values;
$weightedFea_i$: the weighted *fea* by using $termWeightingVec2_i$;
$MicroF_1^i$: result of 10-fold cross validation on $weightedFea_i$;
begin
apply supervised term weighting method to *fea*, and get *termWeightingVec1*;
for $i = 1$ to $|C|$
for $j = 1$ to M
compute $sumVec_i$ for $termWeightingVec1_i$;
end for
end for
sort all $sumVec_i$, and get *sortSum*;
for $i = 1$ to $|C|$
for $j = 1$ to M
for k=1 to i
construct $termWeightingVec2_i$ by the following ways. The *j*-th dimension of each term weighting vector in the selected *k* term weighting vectors is obtained, and the maximum value will be selected as the *j*-th value of the $termWeightingVec2_i$;
end for
end for
end for
for $i = 1$ to $|C|$
compute $weightedFea_i$;
end for
for $i = 1$ to $|C|$
compute $MicroF_1^i$;
end for
record *i* corresponding to the maximum $MicroF_1^i$, and assign it to *selectedC*;
end

In the *ODRS* method, by traversing the term weighting vectors generated by each class, we compare their weighting effects on the training set. The term weighting vector which produces the best effect on training set will be selected as the term weighting vector of test set. We summarize the main process of *ODRS* is shown in the Table 5.

After the algorithm 1 is executed, we can get *selectedC*. Before weighting test set, select the top *selectedC* vector. According to steps 9 to 16, the term weighting vector of test set is constructed.

4. Experimental Results

4.1. Data Corpora

The 20 Newsgroups corpus is a generally used benchmark dataset in the TC [6,7]. In the corpus, there are 20,000 newsgroup documents nearly uniformly distributed into 20 classes. In this paper, we use the 20 Newsgroups sorted by the date. After removing duplicates and headers, the remaining 18,846 documents are partitioned into 11,314 (about 60 percent) training documents and 7,532 (about 40 percent) testing documents. After preprocessing, there are 26,214 distinct words in this data set.

The Reuters21578 corpus is used in many experiments [8,9] and it contains 21,578 documents in 135 categories. We use its ModApte version. There are 5,946 training documents and 2,347 testing documents in this version. In the study, we choose the top 10 largest categories which have 5,228 training documents and 2,057 testing documents. After preprocessing, the resulting vocabulary has 18,221 distinct words.

4.2. Learning Algorithms and Performance Evaluation.

To evaluate classification performance of the proposed method, we choose the promising learning algorithms in this study, i.e., *SVM* classifier [10]. Although other algorithms such as Decision Tree and Naive Bayes are also widely used, they are not included because the real number format of term weights could not be used except for the binary representation (see an exception in [11]).

In this paper, $MicroF_1$ and $MacroF_1$ are employed to measure the performance of the proposed method.

4.3. Experiments.

By taking into account the importance of categories and terms, the optimal document representation is be selected by using *ODRS*.

In order to show the performance of the proposed method, we list the results of optimal *selectedC* which are selected by ODRS. We also report the results of classification experiments with different parameters *selectedC*.

The value of *selectedC* is 20 when *ODRS* is used on 20 Newsgroups datasets which are weighted by *tf***rf* term weighting method.

Table 6 and Table 7 show the $MicroF_1$ and $MacroF_1$ measure result on 20 Newsgroups.

Table 6 A comparison on MicroF1 using ODRS, tf*rf and SVM

selectedC	$MicroF_1$	*selectedC*	$MicroF_1$	*selectedC*	$MicroF_1$
1	0.7562	8	0.7716	15	0.7837
2	0.7592	9	0.7702	16	0.7864
3	0.7631	10	0.7714	17	0.7889
4	0.7629	11	0.7728	18	0.7917
5	0.7634	12	0.7747	19	0.7937
6	0.7631	13	0.7766	**20**	**0.7958**
7	0.7677	14	0.7832		

Table 7 A comparison on MacroF1 using ODRS, tf*rf and SVM

selectedC	$MacroF_1$	*selectedC*	$MacroF_1$	*selectedC*	$MacroF_1$
1	0.7502	8	0.7661	15	0.7786
2	0.7529	9	0.7652	16	0.7823
3	0.7564	10	0.7665	17	0.7841
4	0.7562	11	0.7682	18	0.7876
5	0.7571	12	0.7705	19	0.7893
6	0.7570	13	0.7724	**20**	**0.7909**
7	0.7619	14	0.7785		

The above results indicate that $MicroF_1$ and $MacroF_1$ are similar in values if the same term weighting method is used with different *selectedC*. This is mostly because the 20 Newsgroups corpus contains uniform category.

The value of *selectedC* is 6 when *ODRS* is used on Reuters-21578 datasets which are weighted by *tf***rf* term weighting method.

Table 8 and Table 9 show the $MicroF_1$ and $MacroF_1$ measure result on Reuters-21578.

Table 8 A comparison on MicroF1 using ODRS, tf*rf and SVM

selectedC	$MicroF_1$	*selectedC*	$MicroF_1$	*selectedC*	$MicroF_1$
1	0.9258	5	0.9279	9	0.9269
2	0.9258	**6**	**0.9283**	10	0.9269
3	0.9258	7	0.9272		
4	0.9279	8	0.9269		

Table 9 A comparison on MacroF1 using ODRS, tf*rf and SVM

selectedC	$MacroF_1$	*selectedC*	$MacroF_1$	*selectedC*	$MacroF_1$
1	0.9077	5	0.9033	9	0.9049
2	0.8966	**6**	**0.9086**	10	0.9049
3	0.8957	7	0.9055		
4	0.9021	8	0.9049		

5. Conclusion

In this paper, we have proposed an improved supervised optimal text representation strategy named ODRS, which can search the optimal term weighting vectors on the training set and then applying it on test set. The proposed method is mainly inspired by Younghoong Ko, Yun-Qian Miao and Mohamed Kamel. Younghoong Ko named their model "*D-Max*", "*D-TMax*". Similarly, we can call the proposed optimal model method "*D-NMax*" when ODRS method is used, where *N* is the corresponding value of *selectedC*.

From above results, we get the best result when using "*D-20Max*" on 20Newsgroups. On Reuters-21578, we get the best result when using "*D-6Max*". The main reason for the results is that the distribution of datasets is different. Contrary to 20Newsgroups, the Reuters-21578 is a skewed dataset. In conclusion, the proposed *ODRS* method can find out the effective text representation to improve the classification performance, no matter the dataset is uniform or not.

Acknowledgements

This paper was sponsored by Philosophy and Social Science School Youth Fund Project of Northeast Normal University (Grant No. 19XQ029). We would like to thank the organization for the supports.

References

[1] Lee, Jaesung, et al. "Memetic feature selection for multilabel text categorization using label frequency difference." Information Sciences 485 (2019): 263-280.

[2] Lan M, Tan C L, Su J, et al. Supervised and traditional term weighting methods for automatic text categorization[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 31(4): 721-735 (2009)

[3] Ko, Youngjoong. "A study of term weighting schemes using class information for text classification." Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval. ACM, 2012.

[4] Quan X, Wenyin L, Qiu B. Term weighting schemes for question categorization[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 33(5): 1009-1021 (2011)

[5] Miao, Yun-Qian, and Mohamed Kamel. "Pairwise optimized Rocchio algorithm for text categorization." Pattern Recognition Letters 32.2 (2011): 375-382.

[6] Jiang, Mingyang, et al. "Text classification based on deep belief network and softmax regression." Neural Computing and Applications 29.1 (2018): 61-70.

[7] Lan M, Tan C L, Low H B. Proposing a new term weighting scheme for text categorization[C] AAAI, 6: 763-768 (2006)

[8] Cai, Deng, Xiaofei He, and Jiawei Han, Locally consistent concept factorization for document clustering. Knowledge and Data Engineering, IEEE Transactions on 23:902-913 (2011)

[9] Pereira, Rafael B., et al. "Categorizing feature selection methods for multi-label classification." Artificial Intelligence Review 49.1 (2018): 57-78.

[10] Chang C C, Lin C J. LIBSVM: a library for support vector machines[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2011, 2(3): 27.

[11] Yang, Yiming, An evaluation of statistical approaches to text categorization. Information retrieval 1:69-90 (1999)