Forgetting Word Segmentation in Chinese Text Classification with $L1$-Regularized Logistic Regression*

Qiang Fu, Xinyu Dai**, Shujian Huang, and Jiajun Chen

National Key Laboratory for Novel Software Technology,
Nanjing University, Nanjing 210023, China
{fuq,dxy,huangsj,chenjj}@nlp.nju.edu.cn

Abstract. Word segmentation is commonly a preprocessing step for Chinese text representation in building a text classification system. We have found that Chinese text representation based on segmented words may lose some valuable features for classification, no matter the segmented results are correct or not. To preserve these features, we propose to use character-based N-gram to represent the Chinese text in a larger scale feature space. Considering the sparsity problem of the N-gram data, we suggest the $L1$-regularized logistic regression ($L1$-LR) model to classify Chinese text for better generalization and interpretation. The experimental results demonstrate our proposed method can get better performance than those state-of-the-art methods. Further qualitative analysis also shows that character-based N-gram representation with $L1$-LR is reasonable and effective for text classification.

Keywords: Text classification , Text representation , Chinese Character-based N-gram , $L1$-regularized logistic regression.

1 Introduction

Text classification is a task which automatically assigns an appropriate category for a text according to its content. The task formally can be defined as follows. We have a set of training pairs as $\{(d_1, l_1), (d_2, l_2), ..., (d_n, l_n)\}$, where d_i indicates a text and l_i is the corresponding label drawn from a set of discrete values indexed by $\{1, 2, ..., k\}$. The training data is used to build a classification model h. Then for a given test text t whose class label is unknown, the training model is used to predict the class label l for this text. In recent years, with the rapid explosion of information, text in digital form comes from everywhere. In order to handle a large amount of text, automatically text classification has become not only an important research area, but also a urgent need in different kinds of applications.

* National Natural Science Foundation of China (No. 61003112, 61073119), the National Fundamental Research Program of China (2010CB327903), and the Jiangsu Province Natural Science Foundation(No.BK2011192).

** Corresponding author.

J. Pei et al. (Eds.): PAKDD 2013, Part II, LNAI 7819, pp. 245–255, 2013.

As a result, automatically text classification has been widely studied in machine learning and information retrieval community. When Chinese becomes more and more popular these years, Chinese text classification will be an important way to deal with the large amount of Chinese text.

The common procedure of text classification can be divided into three parts, text representation, feature selection and classification. Generally speaking, before using a classifier, we should present each text as a vector in a high dimensional Euclidean space. Commonly, each word or character in text can be viewed as a feature. And all of these features compose of a feature space. Because the feature space is too large and redundant, for better generalization and performance, we usually apply some dimension reduction or feature selection methods to represent the data in a reasonable scale feature space. The last step is to build a classification model which will be used to predict the class label given new text.

Chinese text classification is a little different from English text classification. We usually need one more step before the common procedure—word segmentation, because Chinese sentences do not delimit words by spaces. Though word segmentation seems a necessary step, we think it may bring some potential problems for classification. Obviously, segmentation errors may bring bad influence to the classification. And even the segmented results are totally correct, some useful information may also be lost which incarnate in our experiments later. Another problem is that word segmentation cannot recognize new words very well. For example, the word-based classification performance is not so good in Social network message data where so many new words exists [7]. So, a natural idea is to use character-based N-gram instead of words.

In this paper, we propose a framework that adopts a character-based N-gram approach to Chinese text classification and uses regularized logistic regression classifier to solve the sparse problem of the N-gram data. There are two main contributions of our work. Firstly, we demonstrate words segmentation will lose some valuable information for classification, no matter the segmented results are correct or not. We also show that character-based N-gram text representation is more suitable for Chinese text classification. Secondly, for better generalization and interpretation, we introduce the $L1$ regularized logistic regression ($L1$-LR) model to classify Chinese text which can get better classification performance.

The rest of this paper is organized as follows. In the next section, we will discuss the background. In section 3, we discuss our works on Chinese text classification, including the detail of proposed classification method and analysis of doing so. In section 4, we present several experimental results and qualitative analysis of N-gram based regularized logistic regression in Chinese text classification followed by conclusions in section 5.

2 Background

In this section, we review some basic information of Chinese text classification, including text representation, feature selection and classifier. In Chinese text representation, one way is to use word segmentation. For Chinese word segmentation, there are two mostly used word segmentation tools: ICTCLAS (Institute

of Computing Technology, Chinese Lexical Analysis System) [2] and Stanford Word Segmenter [3]. Additionally, Stanford Word Segmenter has two certain specifications, that is PeKing University criterion (pku) and Chinese Treebank criterion (ctb). Different criterions will produce different results. Another way to represent text is N-gram based approach, which William and John [5] firstly used in English text classification and received good effects. After preprocessing of text content, we should turn the text into feature vectors next. The commonly used approach is $tf \cdot idf$ [6], which is a weighted technology for text representation. And a high value means the feature (word, character etc.) is important for one text in corpus. Another way is to use 0-1 weight vector, indicating whether one feature appearance in a document or not.

After the generation of the feature vectors, feature selection methods will be used to reduce the feature space. In text classification, commonly used feature selection approaches are Gini Index, Information Gain, Mutual Information and χ^2-Statistic [4]. All of these four methods aim to find the relationship between features and class labels. According to some criterion, these methods give each feature a value that indicate the importance of this feature in classification. But these methods only pay attention to the relation between features and labels, and ignore the relationship between the features which is also important to classification.

After the feature selection is performed, we can use it to training a classifier. In text classification, commonly used classifier are SVM, Logistic regression, Decision Tree, Naive Bayes Classifiers and Neural Network Classifiers. Among these, SVM and Logistic regression are basically linear classifier and do well in text classification. Compared with SVM, the loss function of logistic regression is closer to a linear classifier. So, when we add a regularization term to the classifier, logistic regression is more able to reflect the difference among the difference regularized term than SVM. Furthermore, in large-scale sparse case, logistic regression can perform well compared with SVM [8]. Besides, logistic regression does well in many natural language processing applications [9]. In this paper, we will use logistic regression as our classifier.

3 N-Gram Based Regularized Logistic Regression

Automatic word segmentation error may influence the performance of Chinese text classification. Even if the segmented results are totally correct, some useful information will also be lost. Luo and Ohyama [1] have studied the impact of word segmentation on Chinese text classification. They compared text classification performance on automatic word segmentation, manual word segmentation and character-based N-gram approach, respectively. And they used support vector machine (SVM) with linear kernel, polynomial kernel and radial basis function as classifiers, respectively. The results show that the manual word segmentation gets the best performance, and character-based N-gram also gets the better performance than automatic word segmentation. But in real application, it's almost impossible to get manual word segmentation for each text. From their

work, they don't explain why N-gram features work or not, and what kinds of N-gram feature are most valuable for classification. In this paper, we will totally forget word segmentation and focus on how to effectively extract valuable N-gram features for classification.

Moreover, Zhang and OLES [10] have compared performance of several linear model in text classification, including regularized logistic regression. The results show that regularized logistic regression has a performance that is comparable with other state-of-the-art methods, especially when we have a large scale feature space. And as is well-known, $L1$-regularized logistic regression is an outstanding method to generate a sparse model for classification. The sparse model can give us a chance to dig the huge potential of character-based N-gram for classification. Our work can be divided into two parts. First, we use character-based N-gram approach to represent Chinese text classification. Second, we use regularized logistic regression to train Chinese text classifier.

3.1 Text Representation

Using character-based N-gram to represent text are dramatically simpler, we could avoid the complicated process for the word segmentation. We just extract a sequence of N consecutive characters. In practice, we usually use $N \leq 3$. Table. 1 shows an example of N-gram features.

Table 1. An example of N-gram features

Original Text	南京市长江大桥
1-gram	南;京;市;长;江;大;桥
2-gram	南京;京市;市长;长江;江大;大桥
3-gram	南京市;京市长;市长江;长江大;江大桥

We use $tf{\cdot}idf$ as the feature weight. A $tf{\cdot}idf$ value consists of two parts. One is term frequency (tf), another is inverse document frequency (idf). Term frequency counts the relevant frequency of one feature in a given text. Higher $tf \cdot idf$ value of one feature means that it is more important than others. Inverse document frequency indicates whether one feature appears in most of the documents or not. If one feature appears in most of the documents, it is useless for classification and its idf value will be lower. A $tf \cdot idf$ value for one feature in a text is computed as follows:

$$\frac{n_{i,j}}{\sum\limits_{k} n_{k,j}} \times \log \frac{|D|}{|\{j : t_i \in d_j\}|} \tag{1}$$

where $n_{i,j}$ is the frequency of feature t_i in text d_j, $|D|$ is the total number of documents.

By using N-gram, we can transfer text into feature vectors easily, without any complex word segmentation or other language specific techniques. So, the main measures we adopt are as follows: We use unigram and bigram or unigram, bigram and trigram to represent the text. And $tf \cdot idf$ is used as the weight.

3.2 Regularized Logistic Regression

In logistic regression [10], we model the conditional probability as follows:

$$P(y=1|w,x) = \frac{1}{1+e^{-w^T x}} \tag{2}$$

$$P(y=-1|w,x) = \frac{e^{-w^T x}}{1+e^{-w^T x}} \tag{3}$$

Commonly, we use maximum likelihood estimation (MLE) to obtain an estimate of w, which minimizes the following equation:

$$w = \arg\min_w \sum_{i=1}^{N} \log\left(1+e^{-y_i w^T x_i}\right) \tag{4}$$

Equation 4 may be ill-conditioned numerically. One way to solve this problem is to use regularization. In regularized approximation, by adding a regularizer to the loss function, it can limit model complexity and tune parameters for better generalization. General form of regularization is as follows:

$$\min_f \frac{1}{N} \sum_{i=1}^{N} L(y_i, f(x_i)) + \lambda R(f) \tag{5}$$

Where $L(y_i, f(x_i))$ is the loss function, and $R(f)$ is the regularizer, $\lambda \geq 0$ is a coefficient, which aims to adjust the relationship between the two terms. The goal is to find a model f that is uncomplicated (if possible), and also makes the loss function L small.

In this paper, we use logistic regression with 1-norm regularizer for classification, denote as $L1$-regularized ($L1$-LR) logistic regression. And for comparing, we also use logistic regression with 2-norm regularizer for classification, denote as $L2$-regularized ($L2$-LR) logistic regression. The objective function of $L1$-LR and $L2$-LR are shown as follows, respectively:

$$\min_w \|w\|_1 + C \sum_{i=1}^{N} \log\left(1+e^{-y_i w^T x_i}\right) \tag{6}$$

$$\min_w \frac{1}{2}\|w\|_2 + C \sum_{i=1}^{N} \log\left(1+e^{-y_i w^T x_i}\right) \tag{7}$$

3.3 Analysis of Using $L1$-Regularized Logistic Regression

As mentioned before, we choose $L1$-regularized logistic regression as the classifier. The first reason is that, in character-based N-gram case, the feature space is too large to analysis. And the data sparsity is very serious. The second reason is that most of text classification tasks are linear separable [14]. A simple linear model may perform well compared to complicated models. When compared

with SVM, the loss function of logistic regression is closer to a linear classifier. If we plus a regularized term to the classifier, logistic regression is more able to reflect the difference among the difference regularized term than SVM. Therefore, we choose to use regularizer logistic regression as classifier, especially, the $L1$-regularized logistic regression. Since $L1$-regularization is a sparse model, the feature vector produced by $L1$-regularization has fewer none-zero features. Due to the properties of 1-norm and 2-norm when searching in the hypothesis space, $L1$-regularization encourage less features which may be important in classification to be nonzero and the rest features are zero. On the other hand, $L2$-regularization is more like a kind of average, it encourage more features to be a small value. With this property, $L1$-regularized logistic regression will select some key features from N-gram based feature space. These selected features may seem a bit weird by human, but are definitely valuable for classification. Those selected features can help to interpret the significance of character-based N-gram approach.

Moreover, Andrew Y. Ng [11] proved that using $L1$-regularization, the sample complexity (i.e., the number of training examples required to learn well) grows only logarithmically in the number of irrelevant features. But any rotationally invariant algorithm (e.g., $L2$-regularized logistic regression) exist a worst case that the sample complexity grows at least linearly in the number of irrelevant features. According to this theorem when the irrelevant features are much more than the training text, $L1$-regularization will also achieve a good results. Moreover, text classification is the case that much more irrelevant features come from limited amount of text.

4 Experiments and Results

4.1 Setup

To compare with the previous Chinese text classification, we also implement some common approaches in this paper. As mentioned above, we use ICTCLAS and Stanford Word Segmenter (pku and ctb) as word segmenter, respectively. Top 80 percent word features are selected with four feature selection methods, Gini Index, Information Gain, Mutual Information and χ^2-Statistic. Then, we use SVM as a baseline which is a state-of-the-art classifier in text classification. We use libsvm [12] as tools to train a SVM classifier.

In N-gram based text classification, we use $(1 + 2)$-gram (use unigram and bigram for text representation) and $(1 + 2 + 3)$-gram (use unigram, bigram and trigram for text representation) in experiment. And we use liblinear [13] as tools to train the regularized logistic regression. For solving $L1$-LR, the liblinear use newGLMNET algorithm, see [15] for computational complexity and other details. Additionally, we also use regularized logistic regression on segmented text. At last, we use 10-fold cross validation in our experiments.

4.2 Experiment on Chinese Corpus

Fudan Chinese text classification corpus was used in our experiment (it is released on `http://www.nlpir.org/download/tc-corpus-answer.rar`). We select 9 classes from this corpus. The total number of documents is 9330. The categories include art, history, space, computer, environment, agriculture, economy, politics and sports. For a better comparison with these three different word segmenter, we list the results of N-gram three times. Table. 2 shows the result of text classification on Fudan corpus. Where the $X2$ means that use some word segmenter and χ^2-Statistic as feature selection. The *Gini* means that use some word segmenter and Gini Index as feature selection. The IG means that use some word segmenter and Information Gain as feature selection. The MI means that use some word segmenter and Mutual Information as feature selection. The $L1$-LR means the $L1$-regularized logistic regression. And $L2$-LR means the $L2$-regularized logistic regression.

Table 2. Results of text classification on Fudan corpus

	N-gram		Stanford Word Segmenter(pku)			
	1+2 gram	1+2+3 gram	X2	Gini	IG	MI
$L1$-LR	95.44	**95.57**	94.84	92.27	89.49	86.75
$L2$-LR	95.34	95.31	95.31	92.90	92.52	88.55
SVM	95.35	95.38	95.08	89.48	87.72	81.20
	N-gram		Stanford Word Segmenter(ctb)			
	1+2 gram	1+2+3 gram	X2	Gini	IG	MI
$L1$-LR	95.44	**95.57**	94.88	92.09	88.93	86.79
$L2$-LR	95.34	95.31	95.21	93.00	92.65	88.47
SVM	95.35	95.38	95.16	86.93	86.40	81.16
	N-gram		ICTCLAS			
	1+2 gram	1+2+3 gram	X2	Gini	IG	MI
$L1$-LR	95.44	**95.57**	94.81	92.40	89.43	87.37
$L2$-LR	95.34	95.31	95.29	92.14	91.58	88.10
SVM	95.35	95.38	95.24	88.70	87.20	80.90

From Table. 2, it is obvious that character-based N-gram with $L1$-regularized logistic regression does best in Chinese text classification regardless of which word segmenter is used. Our regularized classifier really does better than traditional features selections methods. And in large scale data classification, regularized logistic regression is more effective than SVM. Additionally, the result shows that dealing with the large and sparse text data, $L1$-regularized logistic regression is better than $L2$-regularized logistic regression.

4.3 Experiment on English Corpus

Moreover, in order to further validate the generality of N-gram based regularized logistic regression approach. We experiment this method on English text

classification. 20-News English text classification corpus was used in our experiment (it is released on `http://qwone.com/~jason/20Newsgroups/`). We use 10 classes selected from 20-News corpus. The total number of documents is 10000 (1000 documents for each class). Then, we repeat the previous steps. Note that for English text, we use word-based N-gram instead of character-based N-gram. The results are shown in Table. 3.

Table 3. Results of text classification on 20-News corpus

	1+2 gram	1+2+3 gram	X2	Gini	IG	MI
$L1$-LR	93.22	**93.73**	92.01	90.94	86.02	83.00
$L2$-LR	91.87	91.87	91.69	91.93	87.26	84.72
SVM	92.05	92.21	91.63	89.99	84.26	81.01

From Table. 3, it is obvious that word-based N-gram with regularized logistic regression does best in English text classification. The result shows that N-gram based regularized logistic regression also performs better than the state-of-the-art approach, in English text classification.

4.4 Accuracy Changes over Nonzero Features

Furthermore, we present the variety curve of text classification accuracy over the number of the nonzero features. We use Fudan corpus as training data and use spline interpolation to reflect the variation trend. The result is shown in Figure. 1. Note that we omit the curve of $(1+2+3)$-gram, since they are almost the same.

In Figure. 1, we can find that text classification accuracy grow rapidly with rising of nonzero features and reach the maximum at around 2000 nonzero features. Then, as the nonzero features continued to increase, the accuracy comes down slightly. This also shows that the sparsity of text data from another side. And a small number of features selected by sparse model are enough to achieve good classification accuracy.

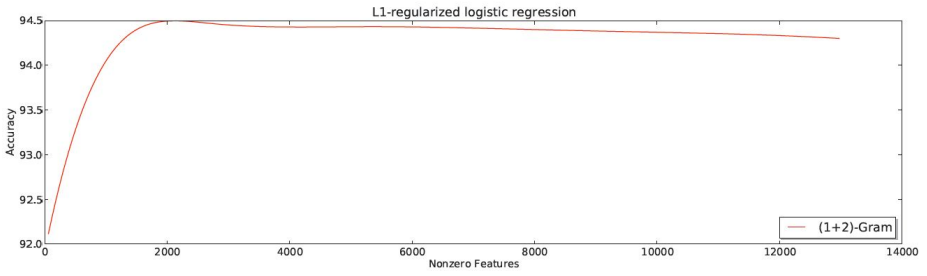

Fig. 1. Accuracy changes over nonzero features

4.5 Qualitative Analysis

At last, in order to qualitative analysis that N-gram based $L1$-regularized logistic regression can select some key features which we cannot obtain through word segmentation. We experiment on a binary-class Chinese text classification problem. We select two class from Fudan Chinese corpus, which composed by 1357 documents labeled as 'computer' and 1601 documents labeled as 'economy'. In order to reflect the importance of each feature more directly, we use 0-1 vector instead of $tf \cdot idf$. Thus, the importance of one feature is totally depend on the weight vector w. Then, we use $(1+2+3)$-gram and $L1$-regularized logistic regression. After the classifier has been trained, we select some typical features from top ranked features. These features are shown in Table. 4.

Table 4. An example of top ranked training features. #occur in Comp is the number of occurrences of the given ngram in documents labeled as 'Computer'. #doc in Comp is the number of documents labeled as 'Computer' which contain the given ngram. #occur in Econ is the number of occurrences of the given ngram in documents labeled as 'Economy'. #doc in Econ is the number of documents labeled as 'Economy' which contain the given ngram.

	Total	#occur in Comp	#doc in Comp	#occur in Econ	#doc in Econ
向对象	757	757	137	0	0
分类名	1489	0	0	1489	1488
o.	1866	1761	1296	105	44
化学报	697	697	511	0	0
经济发	7866	16	11	7850	1220
【原	5556	0	0	5556	1477
原刊地	1275	0	0	1275	1274
政	42714	207	80	42507	1453
识经	3870	12	6	3858	296
期V	752	752	752	0	0
” 。	3710	122	68	3588	1028
主义市	2395	0	0	2395	489

From Table. 4 we can find that most of the top features are N-gram form which cannot be generated by word segmenter. These N-gram features can be divided into three categories. The first category of features presents two separate words. These two words will appear in two classes of text, respectively. But, in one of the two classes, they appear sequentially. Taking '向对象' as an example, this trigram is the suffix of '面向对象' (object-oriented). In segmented text, this feature is segmented as '面向' (oriented) and '对象' (object). The feature '面向' (oriented) appears in 264 computer documents and 230 economy documents, respectively. The feature '对象' (object) appears in 476 computer documents and 406 economy documents, respectively. The number of times they appear in these two classes are similar. As a result, these two features are not useful in classifying the two classes. But the feature '向对象' appears in 137 computer documents and 0 economy documents, respectively. Obviously, this trigram is

more useful than '面向' (oriented) and '对象' (object) for classifying the two classes, significantly. Take '经济发' as another example which is the prefix of '经济发展' (economic development). In segmented text, this feature is segmented as '经济' (economic) and '发展' (development). The feature ' 经济' (economic) appears in 155 computer documents and 1548 economy documents, respectively. The feature '发展' (development) appears in 534 computer documents and 1500 economy documents, respectively. But the feature '经济发' appears in 11 computer documents and 1220 economy documents, respectively, which is a much stronger indicator for classification.

The second category of features consists of only one Chinese character. But this character is usually a prefix or suffix of a group of words. This group of words usually appears in only one of the two classes. Using just one Chinese character to represent a group of words is more effective. Take '政' as an example. In economy text, there are lots of words containing the character '政'. (such as, '政治' (politics), '政策' (policy), '政府' (government), etc.) But, from Table. 4, the feature '政' appears in 80 computer documents and 1453 economy documents, respectively. The number of times they appear in these two classes are different very much. It is obviously that one character will suffice.

The third category of features presents the beginning or end of one sentence. Take '" 。' as an example. In segmented text, they are segmented as '" ' and '。'. But, from Table. 4, the feature '" 。' appears in 68 computer documents and 1028 economy documents, respectively. It is obviously that '" 。' is useful in classification. As an explanation, we find that, in economy text, there are lots of quotes in the end of the sentence.

The above analysis shows that N-gram based $L1$-regularized logistic regression can get some key features which cannot be generated by word segmenter. But these information are useful in text classification, indeed.

5 Conclusions

In this paper, we demonstrate the drawbacks of using word segmentation in Chinese text classification. We propose a framework that use character-based N-gram with regularized logistic regression on Chinese text classification. And, we made experiments on Fudan Chinese text classification corpus. Results of different word segmenters and different feature selection methods are compared. The proposed method gets the best performance. Though quantitative and qualitative analysis, we further discussed for experiments why the N-gram features work better than word features, and what kinds of N-gram features are valuable for classification.

But there are still some limitations of our method. For example, the regularizer we used doesn't consider the relationship between features. We just use 1-norm or 2-norm as the regularizer. In the future work, we will consider adding structure information in the regularizer for better performance hopefully.

References

1. Luo, X., Ohyama, W., Wakabayashi, T., Kimura, F.: Impact of Word Segmentation Errors on Automatic Chinese Text Classification. In: 10th IAPR International Workshop on Document Analysis Systems, pp. 271–275 (2012)
2. Zhang, H., Yu, H., Xiong, D., Liu, Q.: HHMM-based Chinese Lexical Analyzer ICTCLAS. In: Proceedings of the Second SIGHAN Workshop on Chinese Language Processing, Sapporo, Japan, pp. 184–187 (2003)
3. Tseng, H., Chang, P., Andrew, G., Jurafsky, D., Manning, C.: A Conditional Random Field Word Segmenter. In: Fourth SIGHAN Workshop on Chinese Language Processing, Jeju Island, Korea, pp. 168–171 (2005)
4. Aggarwal, C.C., Zhai, C.: A survey of text classification algorithms. In: Mining Text Data, pp. 163–213. Springer (2012)
5. Cavnar, W.B., Trenkle, J.M.: Ngram-based text categorization. In: Proceedings of SDAIR 1994, 3rd Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, US, pp. 161–175 (1994)
6. Salton, G., Fox, E.A., Wu, H.: Extended Boolean information retrieval. Communications of the ACM 26(11), 1022–1036 (1983)
7. Phan, X.H., Nguyen, L.M., Horiguchi, S.: Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In: Proceedings of the 17th International Conference on World Wide Web, pp. 91–100. ACM Press, New York (2008)
8. Komarek, P., Moore, A.: Fast Robust Logistic Regression for Large Sparse Datasets with Binary Outputs. Artificial Intelligence and Statistics (2003)
9. Berger, A.L., Della Pietra, S.A., Della Pietra, V.J.: A Maximum Entropy Approach to Natural Language Processing. Computational Linguistics 22(1), 39–71 (1996)
10. Zhang, T., Oles, F.: Text categorization based on regularized linear classification methods. Information Retrieval, 5–31 (2001)
11. Andrew, Y., Ng: Feature selection, l1 vs. l2 regularization, and rotational invariance. In: Proceedings of the Twenty-First International Conference on Machine learning (ICML), pp. 78–85. ACM Press, New York (2004)
12. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27 (2011), `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
13. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: LIBLINEAR: A library for large linear classification. Journal of Machine Learning Research 9, 1871–1874 (2008)
14. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: Proceedings of the 10th European Conference on Machine Learning, Heidelberg, Germany, pp. 137–142 (1998)
15. Yuan, G.X., Ho, C.H., Lin, C.J.: An improved glmnet for l1-regularized logistic regression. The Journal of Machine Learning Research, 1999–2030 (2012)