A Weight-based Feature Extraction Approach for Text Classification

Jung-Yi Jiang and Shie-Jue Lee
Dept. of Electrical Engineering, National Sun Yat-Sen University, Taiwan
{ jungyi|leesj }@water.ee.nsysu.edu.tw

Abstract

In this paper, we propose a weight-based feature extraction approach to reduce the number of features for text classification. The number of extracted features is equal to the number of document classes and the feature values are obtained according to the distributions of words over class partitions. Each word of the original word set contributes a weight to each extracted feature and a transformation matrix is formed. By using the transformation matrix, the original document set is converted to a new set with a smaller number of features. The proposed approach has two advantages. Trial-and-error for determining the appropriate number of extracted features can be avoided. Computation demand is small and the method runs fast. Experimental results obtained from real-world data sets have shown that our method can perform better than other methods.

1. Introduction

In this paper, a new feature reduction approach for document data is proposed. Recently, text data processing approaches have attracted more and more attention. These approaches have to deal with an important problem of a large number of features. For example, two real-world data sets, 20 Newsgroups and Reuters21578 top-10, both have more than 15,000 features. Such high dimensionality is a severe obstacle for classification algorithms [1]. To alleviate this difficulty, feature reduction approaches are applied before document classification tasks are performed.

Two major approaches, feature selection and feature extraction, have been proposed for feature reduction. The feature selection methods select a subset of the original features and the classifier only uses the subset instead of all the original features to perform the text classification task. A well-known feature selection approach is based on Information Gain [3], which is an information-theoretic measure defined by the amount of reduced uncertainty given a piece of information. The feature extraction methods convert the representation of the original documents to a new representation based on a smaller set of synthesized features. Word clustering [4]-[8] is one of effective techniques for feature extraction. The idea of word clustering is to group words with a high degree of pairwise semantic relatedness into clusters and each word cluster is then treated as a single feature and thus feature dimensionality can be drastically reduced.

The first feature extraction method based on word clustering was suggested by Baker and McCallum[4] derived from the 'distributional clustering' idea of Pereira et al. [7]. An Information Bottleneck approach was proposed by Tishby et al. [5][6] and showed that word clustering approaches are more effective than feature selection ones. A Divisive Information-Theoretic method was proposed by Dhillon et al. [8], which is more effective than other word clustering methods. However, both information gain and clustering word based methods only use a part of the original words to generate new features. For information gain based methods, only a subset of the original words is used. For word clustering based method, each new feature is generated by combining a subset of the original words. Such methods ignore useful information that may be provided by the unused words.

In this paper, we propose a weight-based feature extraction approach to reduce the number of features for text classification. The number of extracted features is equal to the number of document classes and the feature values are obtained according to the distributions of words over class partitions. Each word of the original word set contributes a weight to each extracted feature and a transformation matrix is formed. By using the transformation matrix, the original document set is converted to a new set with a smaller number of features. The proposed approach has two advantages. Trial-and-error for determining the appropriate number of extracted features can be avoided. Computation demand is small and the method runs fast. Experimental results obtained from two real-world data sets, 20 Newsgroup and Reuter-21578 Top 10, have shown that our method can perform better than information gain and word clustering based methods.

0-7695-2882-1/07 $25.00 ©2007 IEEE

2. Background and related work

To process documents, the bag-of-words model[2] is usually used. Let d_i be a document and the set $D = \{d_1, d_2, \ldots, d_n\}$ represent n documents. Let the word set $W = \{w_1, w_2, \ldots, w_m\}$ be the feature set of the documents. Each document d_i, $1 \leq i \leq n$, can be represented as $d_i =< w_{i1}, w_{i2}, \ldots, w_{im} >$, where each w_{ij} denotes the number of occurrence of w_j in document d_i. The feature reduction task is to find a new word set $W' = \{w'_1, w'_2, \ldots, w'_k\}$, $k < m$, such that W and W' work equally well for all the desired properties with D. After feature reduction, each document d_i is converted to a new representation $d'_i =< w'_{i1}, w'_{i2}, \ldots, w'_{ik} >$ and the converted document set is D' = $\{d'_1, d'_2, \ldots, d'_n\}$. If k is very much smaller than m, computation cost can be drastically reduced.

2.1. Feature selection

In feature selection approaches[3], the new feature set $W' = \{w'_1, w'_2, \ldots, w'_k\}$ is a subset of the original features. This approach only uses the selected features as inputs for classification tasks.

Information Gain is frequently employed in the feature selection approach. It measures the reduced uncertainty by an information-theoretic measure and gives each word a weight. The bigger the weight of a word is, the larger the reduced uncertainty by the word is. Let $\{c_1, c_2, ..., c_p\}$ denote the set of classes. The weight of a word w_i is calculated as follows:

$$\begin{aligned} G(w_i) = \quad & -\textstyle\sum_{l=1}^{p} Pr(c_l)logPr(c_l) \\ & +Pr(w_i)\textstyle\sum_{l=1}^{p} Pr(c_l|w_i)logPr(c_l|w_i) \\ & +Pr(\overline{w}_i)\textstyle\sum_{l=1}^{p} Pr(c_l|\overline{w}_i)logPr(c_l|\overline{w}_i). \end{aligned} \tag{1}$$

The words of top k weights in W are selected as the features in W'.

2.2. Feature extraction

Unlike feature selection, feature extraction combines the original features to generate new features. For example, the word clustering based feature extraction methods combine the words of a subset of the original features into a new feature.

The word clustering methods proposed in [4]-[8] are “hard” clustering methods where each word of the original features belongs to only one word cluster. Therefore each word contributes to the synthesis of only one new feature. Each new feature is obtained by summing up the words belonging to one cluster. The new feature set $W' = \{w'_1, w'_2, \ldots, w'_k\}$ corresponds to a partition $\{W_1, W_2, \ldots, W_k\}$ of W, i.e., $W_j \bigcap W_q = \emptyset$, where $1 \leq q, j \leq k$ and $j \neq q$. Note that a cluster is equivalent to an element in the partition.

The distributional word clustering method calculates the distributions of words over classes, $Pr(C|w_i)$, $1 \leq i \leq m$, where $C = \{c_1, c_2, ..., c_p\}$ and p is the number of class labels, and uses *Kullback-Leibler* divergence to measure the dissimilarity between two distributions. The distribution of a cluster W_j is calculated as follows:

$$P(C|W_j) = \sum_{w_t \in W_j} \frac{P(w_t)}{\sum_{w_t \in W_j} P(w_t)} P(C|w_t). \tag{2}$$

The goal of distributional word clustering is to minimize the following objective function:

$$\sum_{j=1}^{k} \sum_{w_t \in W_j} P(w_t) KL(P(C|w_t), P(C|W_j)). \tag{3}$$

Which takes the sum over all the clusters.

3. Proposed method

Let D be the matrix consisting of all the original document with m features and D' be the matrix consisting of the converted document with new k features. The feature reduction task can be written in the following form:

$$D' = DT. \tag{4}$$

where

$$D = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix}, D' = \begin{bmatrix} d'_1 \\ d'_2 \\ \vdots \\ d'_n \end{bmatrix}, T = \begin{bmatrix} t_{11} & \ldots & t_{1k} \\ t_{21} & \ldots & t_{1k} \\ \vdots & \ddots & \vdots \\ t_{m1} & \ldots & t_{mk} \end{bmatrix}.$$

Our goal is to find a transformation matrix T to convert D to D' in a desirable way.

The document classification task is a supervised work where each document of the training data set has a given class label. Since we have the information about class labels, intuitively we can synthesize a new feature to distinguish the documents of one class from the documents of the other classes. Also, it seems reasonable that a word is well related to a class if the word occurs more frequently in the documents of the class. By this motivation, we propose a new approach to feature extraction for text classification. Firstly, we assume that we can synthesize new features from the original features to distinguish one class from another, and that the number of new features is equal to the number of classes. Secondly, we assume that new features can be obtained by considering the degrees to which the original words are related to the classes. Based on these ideas, we generate a transformation matrix and each element of the matrix denotes a weight from one of the original features associated with a new feature which corresponds to a certain class. The weight for a word is large if it occurs frequently in the documents of the class with the underlying new feature. On the contrary, the weight is smaller if the word occurs less frequently in the documents of the underlying class. Formally, we define the elements of transformation matrix T in equation(4) as follows:

$$t_{ij} = Pr(c_j|w_i). \tag{5}$$

where

$$Pr(c_j|w_i) = \frac{\text{\# of occurence of } w_i \text{ in class } c_j}{\text{\# of occurence of } w_i \text{ in all classes}}. \tag{6}$$

Thus, each new feature value w'_{rj} of document d_r can be calculated as follows:

$$w'_{rj} = \sum_{i=1}^{m} w_{ri} \times Pr(c_j|w_i). \tag{7}$$

Our method works in a straightforward way. Firstly, we calculate the probabilities of words over classes. Then, we generate the transformation matrix. Finally, we use the transformation matrix to convert documents from the original features to new features. For clarity, the algorithm of our method is summarized below:

```
Input: D is the set of documents, W is the set of words, C
       is the set of classes, l is the number of classes, and
       m is the number of words.
Output: D' is the set of converted documents.

1. For each word $w_i \in W$ and each class $c_j \in C$,
   $1 \leq i \leq m$ and $1 \leq j \leq l$, calculate $Pr(c_j|w_i)$
   by equation (6).
2. Obtain transformation matrix T by equation (5).
3. Convert document set D to new document set D'
   by equation (4).
```

After transformation of documents is done, we can perform the classification task with the converted data instead of the original data. The computation of the transformation matrix is to estimate the conditional probabilities of a class given a word. The probability estimation has a time complexity proportional to the number of documents. Our method has two advantages. Trial-and-error for determining the appropriate number of extracted features can be avoided. Computation demand is small and the method runs fast.

4. Experiments and Results

To show the effectiveness of our proposed method, experiments on two well-known data sets for text classification research, 20 Newsgroup (20NG) and Reuters-21578, are performed. Experiment 1 works on the 20 Newsgroup (20NG) corpus which contains about 20000 articles taken from the Usenet newsgroups. These articles are evenly distributed over 20 categories; each category of 20 Newsgroup has about 1000 articles. We use two-thirds of the documents for training and the rest for testing. The documents of Reuters-21578 are divided, according to the "ModApte" split, into 9603 training documents and 3299 testing documents. To make a difference from 20 Newsgroup, the distribution of documents is skewed. The number of training documents per class varies from 1 to about 4000, with top 10 classes containing 77.5% of the documents and 28 classes have fewer than 10 training documents. Experiment 2 uses the documents of the top 10 classes. The number of words involved in Experiment 1 and Experiment 2 is 25718 and 16285, respectively. To demonstrate the classification capability of the reduced features, we choose the Naive Bayes classifier to do text classification. We compare our method with other methods on the classification accuracy and running speed.

4.1. Experiment 1: 20 Newsgroup Data

Table 1 and table 2 show the classification accuracy (%) and execution time (sec) of the 20 Newsgroup data set obtained by our method, the Divisive Clustering (DC) based feature extraction method, and the Information Gain (IG) based feature selection method, respectively. Note that the 20 Newsgroup data set contains 25718 features.

Accuracy % of our method with **20 features: 88.18**

Method	**Number of features**					
	2	5	10	**20**	50	100
DC	15.88	37.02	54.16	**78.54**	83.6	85.32
IG	4.32	7.79	13.39	**18.34**	28.48	34.71
Method	**Number of features**					
	200	500	1000	5000	25718	
DC	86.80	87.80	88.05	89.20	88.40	
IG	45.38	63.14	73.76	84.65	88.40	

Table 1. Accuracy % of three approaches on 20 Newsgroup data with 1/3-2/3 test-training split.

Execution time (sec) of our method : **1300.8**
Execution time (sec) of IG: **1337.3**

Method	**Number of features**				
	2	5	10	**20**	50
DC	1442.0	1565.1	1698.1	**1817.3**	2178.5
Method	**Number of features**				
	100	200	500	1000	5000
DC	2799.1	4255.5	8896.9	17108.4	47235.6

Table 2. Execution time (sec) of three approaches on 20 Newsgroup data with 1/3-2/3 test-training split.

As shown in these tables, our method achieves 88.18% accuracy with 20 features in 1300.8 seconds. The accuracy is just 0.22% lower than that achieved by a full feature Naive Bayes classifier (88.40%). With 20 features, DC achieves 78.54% accuracy in 1817.3 seconds and IG achieves 18.34% accuracy in 1337.3 seconds. DC is better than ours in accuracy only when the number of features is more than 5000, but it spends much more time than ours. For example, DC achieves 89.2% with 5000 features in 47235.6 seconds. IG is better than ours only when the full features is used. Our method achieves almost the best accuracy DC or IG can achieve, but in much less time.

4.2. Experiment 2: Reuter-21578 Top 10 Data

Table 3 and table 4 show the classification accuracy (%) and execution time (sec) results of the Reuter-21578 Top 10 Data set obtained by our method, the Divisive Clustering (DC) based feature extraction method, and the Information Gain (IG) based feature selection method, respectively. Note that this data set contains 16285 features.

Accuracy % of our method with **10 features: 83.75**

Method	**Number of features**					
	2	5	**10**	20	50	100
DC	49.00	76.50	**80.20**	81.28	83.72	82.89
IG	41.54	52.33	**55.06**	57.35	62.23	68.54
Method	**Number of features**					
	200	500	1000	5000	16285	
DC	83.86	84.47	84.33	84.43	86.27	
IG	72.53	83.72	84.65	86.41	86.27	

Table 3. Accuracy % of three approaches on Reuter-21578 Top 10 data.

Execution time (sec) of our method : **492.1**
Execution time (sec) of Information Gain: **513.6**

Method	**Number of features**				
	2	5	**10**	20	50
DC	518.4	525.3	**563.3**	622.5	854.1
Method	**Number of features**				
	100	200	500	1000	5000
DC	1100.3	1898.9	3418.5	5151.1	17781.2

Table 4. Execution time (sec) of three approaches on Reuter-21578 Top 10 data.

As shown in these tables, our method achieves 83.75% accuracy with 10 features in 492.1 seconds and the accuracy is just 2.52% lower than the accuracy achieved by a full feature Naive Bayes classifier (86.27%). With 10 features, DC achieves 80.2% accuracy in 563.3 seconds and IG achieves 55.06% accuracy in 513.6 seconds. DC is better than ours only when the number of features is more than 200 but spends much more time (more than 1898.9 seconds). DC achieves 84.47% with 500 features in 3814.5 seconds and IG achieves 86.41% with 5000 features in 513.6 seconds. Our method can achieve very good accuracy with much less time than DC and IG.

5. Conclusion

We have proposed a weight-based feature extraction approach for document classification. The number of extracted features is equal to the number of document classes and the feature values are obtained according to the distributions of words over class partitions. The proposed approach has two advantages. Trial-and-error for determining the appropriate number of extracted features can be avoided. Computation demand is small and the method runs fast. Experimental results obtained from two real-world data sets, 20 Newsgroup (20NG) and Reuters-21578, have shown that our method can achieve very good classification accuracy in much less time than the divisive clustering based feature extraction method and the information gain based feature selection method.

References

[1] F. Sebastiani, *Machine Learning in Automated Text Categorization.* ACM Computing Surveys, Vol, 34, No.1, March 2002, pp.1-47.

[2] G. Salton and M. J. McGill, *Introduction to Modern Retrieval.* McGraw-Hill Book Company, 1983.

[3] Y. Yang and J. O. Pedersen, *A comparative study on feature selection in text categorization.* In Proceedings of 14th International Conference on Machine Learning, Morgan Kaufmann, 1997, pp.412-420.

[4] L. D. Baker and A. McCallum, *Distributional clustering of words for text classification.* In Proceedings of 21st Annual International ACM SIGIR, 1998, pp.96-103.

[5] N. Slonim and N. Tishby, *The power of word clusters for text classification.* In Proceedings of 23rd European Colloquium on Information Retrieval Research (ECIR), 2001.

[6] R. Bekkerman, R. El-Yaniv, N. Tishby and Y. Winter, *Distributional Word Clusters vs. Words for Text Categorization.* Journal of Machine Learning Research 1, 2002, pp.1-48.

[7] F. Pereira, N. Tishby and L. Lee, *Distributional clustering of English words.* In 31st Annual Meeting of ACL, 1993, pp.183-190

[8] I. S. Dhillon, S. Mallela and R. Kumar, *A Divisive Infromation-Theoretic Feature Clustering Algorithm for Text Classification.* Journal of Machine Learning Research 3, 2003, pp.1265-1287.