A Text Categorization Method Based on Improved k-means and BP Neural Network

Rongze Xia
School of Computer, National University of Defense Technology
Changsha, 410073, China
e-mail: XiaRongze1986@163.com

Yan Jia
School of Computer, National University of Defense Technology
Changsha, 410073, China
e-mail: jiayanjy@vip.sina.com

Hu Li
School of Computer, National University of Defense Technology
Changsha, 410073, China
e-mail: lihu@nudt.edu.cn

***Abstract*—K-means is a widely used cluster algorithm. It is widely used in text categorization as an unsupervised method. However, it could be easily affected by some isolated observations. BP neural network is usually used for text categorization because it's superiority in handling non-linear problem. However, sometimes it could not achieve high performance. Based on the combination of these two algorithms, we propose a new text categorization algorithm. We first improve k-means clustering algorithm. After that, we use it to cluster vectors in our vector space model. And then, BP neural network is used to categorize the preprocessed vectors. The experiments show that our algorithm could achieve a high performance than the traditional BP neural network text categorization method.**

Keywords-BP neural network; text categorization; k-means

I. INTRODUCTION

With the development of Internet, the information on it increases explosively. We could get a lot of information through it. Under this condition, we have to know to what topic the information is relevant or whether the information is important. Text categorization is a useful tool for us to do above works.

Let's give a description to text categorization problem. It's a task to classify several unlabeled documents into some predefined categories. To classify a document rightly, first we should train a model according to some labeled documents. It needs us to manually label each document. This process is time-consuming. Another solution for text categorization of unlabeled documents is unsupervised learning method. K-means clustering is a common algorithm for this problem. Through k-means, we do not need to pre-label documents for a training model. However, k-means clustering is not accurate enough for text categorization problem. It is sensitive to some noise samples. So we improve traditional k-means clustering algorithm by eliminating noise samples.

Text categorization problem is a high dimension problem. It's suitable for BP neural network (BPNN) to solve. However, BPNN is not interpretable. It could not achieve high performance in text categorization. In this paper, we combined k-means and BPNN. First we use our improved k-means to cluster unlabeled documents. After that, the BPNN is used to classify documents to different categories. Our experiments results show that our method could outperform the traditional BPNN text categorization method.

The following paper proceeds as this. In section 2, we review related work briefly. In section 3, we discuss the improvement of k-means, BPNN and the combination of these two algorithms. In section 4, we show our experiments results. In section 5, we conclude our work.

II. RELATED WORK

There are many research have been done on the text categorization problem. Many clustering algorithms are used in this area. Ding et al. [1] proposed a coherent framework to adaptively select the most discriminative subspace in text categorization. The linear discriminant analysis and k-means clustering algorithm were combined in this framework. K-means clustering algorithm was used to generate class labels and linear discriminant analysis was used for subspace selection. Their experiments showed that their combined framework was better than traditional subspace selection methods such as PCA, LDA and k-means. MODHA et al. [2] presented a framework to integrate multiple feature spaces in text categorization. K-means clustering algorithm was used to build this framework. Their framework could determine the optimal feature weighting. At the same time, the average within-cluster dispersion could be minimized while the average between-cluster dispersion could be maximized. Experiments results demonstrated the high effectiveness of their framework. Boley et al. [3] proposed two new clustering techniques for high dimensional feature space reduction in text categorization. These clustering techniques were based on the generalization of graph partitioning. Several experiments which were conducted could achieve high performance. Almeida et al. [4] proposed a SVM-KM algorithm. This algorithm was based on k-means clustering algorithm and Support Vector Machine. When the vectors were in the SVM optimization phase, they could be preserved or not as support vectors based on the distances between them and separation margins. When the vectors were in the k-means phase, they could be grouped in many clusters. The training time of SVM-KM algorithm was decreased without compromising the generalization capability of SVM.

Neural network have also been widely used in text categorization. Chau et al. [5] proposed an intelligent method for enabling concept-based hierarchical multilingual text categorization. Through their method, a universal concept space was constructed and a set of concept-based multilingual document categories were generated. Neural network in this method was used for building a concept-based multilingual text classifier. Souza et al. [6] examined virtual generalizing random access memory weightless neural network. They used this neural network to build an automatic multi-label text categorization system. The test on categorization of free-text descriptions of economic activities and web pages showed that their method could achieve high performance. Zhang et al. [7] proposed a neural network algorithm which was called BP-MLL. It was designed for Multi-Label Learning. A novel error function was employed for capturing the characteristics of multi-label learning. The real experiments on text categorization showed that the performance of BP-MLL was superior to other multi-label learning algorithms. Jo [8] proposed a neural network model, which was called “NeuroTextCategorizer”. It was used for content based text categorization. This model solved the main problems of traditional model, which were the high dimensionality and sparse distributions of vectors.

III. Text Categorization Method Based on K-means and BP Neural Network

In this section, we give a description of our text categorization method. Our method is based on improved k-means and BPNN. We will first introduce vector space model because in our method every document is represented as a vector.

A. *Vector Space Model*

Vector space model is the most widely used method in the representation of documents. In this model, every document is represented as a vector. For a document d, we could represent every word in d as a term t, so document d could be expressed as a vector $d = < t_1, t_2, \cdots, t_m >$. t_i is the *i*-th term of the vector. The length of the vector is m, that means there are m terms in document d. Different terms in document d play different roles. Some of them are more relevant to the overall topic; some of them are less relevant to that. We assign different weights to these terms according to their importance. So the vector could be expressed as this: $d = < w_1, w_2, \cdots, w_m >$. w_i is the weight of the *i*-th term.

The conventional method for calculating the term weight is TF-IDF. Here we use TF-IDF in our method. TF is short for term frequency. It’s an importance measurement of a term in the document. When a term appears more times in a document, it’s more relevant to the subject of the document. And the TF value is higher. IDF is short for inverse document frequency. It’s an importance measurement of a term in the whole corpus. If a term appears more times in the whole corpus, that means many documents could contain this term. The distinguish capacity of this term is lower.

B. *Improved K-means Clustering Algorithm*

K-means clustering is a common clustering algorithm. *k* is the parameter that we should pre-define. It could partition *n* observations into *k* clusters. First, *k* cluster center is selected. And then, every observation is assigned to the cluster center which is closest to it. And then, the cluster center is updated to be the mean of observations in the cluster. The above operations will be done again and again until cluster centers do not change.

However, the traditional k-means clustering algorithm is sensitive to some noise observations. The updating of cluster center is based on the mean of observations in the cluster. Once a noise observation is isolated, it could affect the generation of new cluster center. In this paper, we improved traditional k-means clustering algorithm. We improve k-means performance by eliminating those isolated observations. The improving k-means algorithm will be described as following:

1) Eliminating isolated observations: Supppose there are a total *n* observations. these observations could be expressed as $< t_1, t_2, \cdots, t_n >$. For $t_i (i = 1, 2, \cdots, n)$ we calculate the distance sum between t_i and other $n-1$ observations. We identify this distance sum of t_i as $dist_i$. Then we will get a vector $< dist_1, dist_2, \cdots, dist_n >$. If $dist_i$ is as 3 times as the average value of other $n-1$ element, we think observation *i* is an isolated observation, so we eliminate it.

2) Initialization: we randomly select *k* observations as cluster centers. They are identified as $< m_1, m_2, \cdots, m_k >$.

3) Assign observations into clusters: we calculate the distance between each observation x and each cluster center m_j. If $\| x - m_i \| < \| x - m_j \| (i, j = 1, 2, \cdots, k, i \neq j)$, then we assign the observation x into the cluster *i* because center m_i is closest to it.

4) Updating cluster center: When every observation is assigned to a cluster, we should update the cluster center. The new cluster center is the new means of all observations in this cluster. We could calculate the mean through the following equation:

$$m_i = \frac{1}{| S_i |} \sum_{x_j \in s_i} x_j (j = 1, 2, \cdots, | S_i |) \tag{1}$$

In the above equation, m_i is the mean of the i-th cluster, which is indicated by S_i. x_j is the j-th observations in cluster S_i.

C. BP Neural Network

BPNN is a multi-layer feed-forward network. There are usually three layers in this network. The first layer is an input layer for receiving input. The second layer is a hidden layer. The third layer is an output layer for output results. There are many neurons and connections in the BPNN. The source of a connection is a neuron in the previous layer and the end of a connection is a neuron in the succeeding layer. Each connection will be assigned a weight. The number of neurons in the input layer is equal to the vector length in the vector space model. That means vectors are the input of the BPNN. The number of neurons in the output layer is equal to the categories number. Each neuron in this layer represents a category. We should notice that the hidden layer has great impact on the generalization ability of BPNN. It's important to determine the neurons number of this layer. There are many ways for the determination of this. We use the Baum-Haussler rule [9] to set the number of hidden layer neurons.

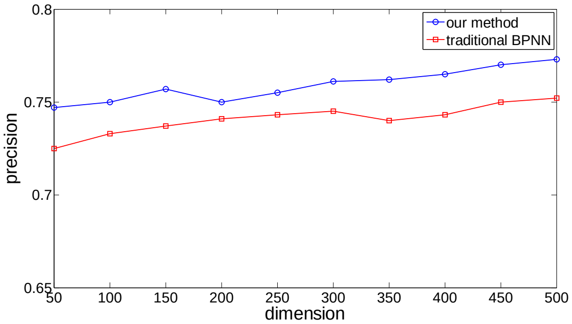

Figure 1. Precision comparison of two methods

$$p \leq (N \cdot E)/(m+n) \tag{2}$$

In the above equation, p is the number of neurons in the hidden layer. N is the number of training examples. E is the error tolerance. m and n are the number of neurons in the input layer and output layer respectively.

The BPNN text categorization method could be describe as following: we input a vector which represents a document into BPNN. Because the neuron number of input layer is equal to the vector length, every neuron in the input layer receives a term in the vector. After that, BPNN will be trained. Finally, it will output a vector $V = < v_1, v_2, \cdots, v_n >$. In this vector, if the maximum value $v_{\max}$ is the i-th one in the vector V, then we should know that the input document should be classified to the i-th category.

The training process of a BPNN is as this: In the training step the network will calculate based on the input vector and output a result. If the result is not correct, the BPNN will back propagate to update weights of connections. And BPNN will calculate and output a result again based on the updated connection weights. The above process will repeat continuously until we get the right result.

D. A Combination Of K-Means And BP Neural Network

The traditional BPNN text categorization method could not achieve high performance. So in this paper, we combine our improved k-means cluster algorithm and BPNN. We use improved k-means algorithm to cluster texts into some categories. This is at a coarse-grained level. After that, we use BPNN for text categorization at a fine-grained level. The steps are as following:

1) Clustering text: In this step, we should use our improved k-means algorithm to cluster all the unlabeled texts into k cluters. Each cluster represents a category.

2) Training BPNN: Once we get k clusters, we could use observations in them to train a BPNN. For every cluster, we pick those texts which is close to the cluster center as our training samples.

3) Text Categorization: When we get a well-trained BPNN, we could use this BPNN for text categorization problem.

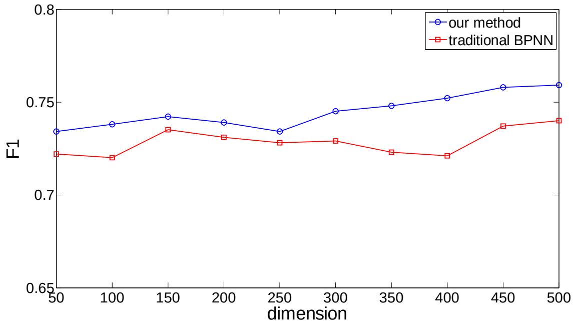

Figure 2. F1 value comparison of two methods

IV. Experiment And Analysis

A. Datasets And Prepared Work

Fudan University's natural process language corpus are used in experiments. This corpus could be access by anyone. It's often used for research especially in text categorization area. We use half of the whole corpus in our experiments. There are 5930 Chinese documents. These documents are pre-divided into 10 categories, which are art, literature, education and so on. However, in our experiment, we will not take their categories into account. We treat all these texts as unlabeled texts. They are split into training sets and test sets. There are 2963 documents in the training sets and 2967 documents in the test sets. We do Chinese word segmentation by ICTCLAS [10], which is a widely used Chinese word segmentation tool. Weka [11] is another important tool for our experiments. We use it to build a vector space model and reduce the feature space.

There are many widely used criteria to evaluate the performance of a text categorization method. In this paper, we use two traditional measurements: one is precision, the other one is F1 value. At the same time, we take the run time as a measurement into account. It could evaluate the efficiency of text categorization methods. We evaluate these different measurements on condition of different vector dimensions.

B. Result Analysis

We compare the precision of two methods. The results are showed in Fig. 1. From Fig. 1, we could know that when the dimension of vectors increases, the precision of two methods keeps increasing. For the precision of traditional BPNN, we know that the lowest precision is 0.725. It increases to its first peak, which is 0.745 when the dimension is 300. When the dimension is 350, it reaches to a lower value which is 0.741. After that, the precision increases again and reaches to the highest value, which is 0.752. The precision of our method is higher than that of traditional BPNN. The overall trend continues to increase. It increases from the lowest value 0.740 to the highest value, which is 0.773 when the dimension is 500. There is only one exception, which is when the dimension is 200, it decreases to 0.750. The average precision of our method is 0.759, which is higher than 0.741, the average value of traditional BPNN precision.

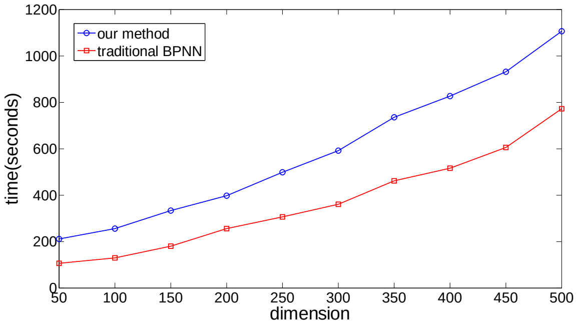

Figure 3. Run time comparison of two methods

The comparison of F1 value between traditional BPNN and our method are showed in Fig. 2. Let's investigate them respectively. The F1 value of traditional BPNN is not very regular. Its lowest value is 0.720, when the dimension is 100. It reaches the first peak when the value is 150. The first peak value is 0.735. After that, this F1 value slowly decreases. When the dimension is 400, the F1 value is 0.721. After that, it increases rapidly. The highest value is 0.740 when dimension is 500. However, The F1 value trend of our method is more stable than that of traditional BPNN. It always keeps increasing with the increase of dimension with one exception. That is when the dimension is 200 and 250, the F1 value 0.739 and 0.734 respectively. It reaches the highest value which is 0.759 when the dimension is 500. For our method, the average F1 value is 0.745. It's higher than that of traditional BPNN, which is 0.727.

Now let's pay attention to the run time of these two methods. We could clearly see the difference between them. The run time curve of traditional BPNN is lower than that of our method. When the dimension is 50, the run time needs only 105 seconds. It increases relatively much slower than the run time of our method. When dimension is 500, it takes 771 seconds to complete the text categorization. However, the run time of our method takes much more time. When the dimension is 50, it takes 211 seconds. After that, it increases nearly exponentially. When the dimension increases to 500, the run time of our method is almost 1105 seconds. Because it takes a lot of time to eliminate isolated observations.

V. Conclusion

A lot of work has been done one text categorization based on neural network. Traditional BPNN in text categorization problem could not achieve high performance. In this paper, we combine k-means clustering algorithm and BPNN for text categorization. At the same time, we improve k-means clustering algorithm by eliminating isolated observations to increase its performance. Experiments results show that our method could achieve higher performance than traditional BPNN method.

Acknowledgment

The authors' work was sponsored by China's 973 Program(2013CB329601, 2013CB329602), National High Technology Research and Development Program (863) of China(2011AA010702, 2012AA01A401 and 2012AA01A402), the Nature Science Foundation of China (60933005, 91124002), Support Science and Technology Project of China (2012BAH38B04, 2012BAH38B06), China Postdoctoral Science Foundation Program(2012M520114).

References

[1] Ding, C. and T. Li. "Adaptive dimension reduction using discriminant analysis and k-means clustering." ACM International Conference Proceeding Series.

[2] Modha, D. S. and W. S. Spangler. "Feature weighting in k-means clustering." Machine learning 52(3): 217-237.

[3] Boley, D., M. Gini, et al. "Partitioning-based clustering for web document categorization." Decision Support Systems 27(3): 329-341.

[4] Barros de Almeida, M., A. de P ádua Braga, et al. "SVM-KM: speeding SVMs learning with a priori cluster selection and k-means." Neural Networks, 2000. Proceedings. Sixth Brazilian Symposium on, IEEE.

[5] Chau, R., C. Yeh, K. A. Smith, "A neural network model for hierarchical multilingual text categorization," Advances in Neural Networks–ISNN 2005, Springer: 238-245.

[6] De Souza, A. F., F. Pedroni, et al. "Automated multi-label text categorization with VG-RAM weightless neural networks," Neurocomputing 72(10): 2209-2217.

[7] Zhang, M., Z. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," Knowledge and Data Engineering, IEEE Transactions on 18(10): 1338-1351.

[8] Jo, T., "Neurotextcategorizer: A new model of neural network for text categorization," Proceedings of the International Conference of Neural Information Processing.

[9] Baum, E. B., D. Haussler, "What size net gives valid generalization," J. Neural Computation. 1, 1, 151-160

[10] http:// www.ictclas.org

[11] http://weka.wikispaces.com