VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui

School of Mathematics, Statistics and Computer Science
Computer Science

Scientific Paper Classification using Neural Networks

Mengjie Zhang, Xiaoying Gao, Minh Duc Cao

Technical Report CS-TR-06/9
March 2006

School of Mathematics, Statistics and Computer Science
Victoria University
PO Box 600, Wellington
New Zealand

Tel: +64 4 463 5341
Fax: +64 4 463 5045
Email: Tech.Reports@mcs.vuw.ac.nz
http://www.mcs.vuw.ac.nz/research

VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui

School of Mathematics, Statistics and Computer Science
Computer Science

PO Box 600
Wellington
New Zealand

Tel: +64 4 463 5341, Fax: +64 4 463 5045
Email: Tech.Reports@mcs.vuw.ac.nz
http://www.mcs.vuw.ac.nz/research

Scientific Paper Classification using Neural Networks

Mengjie Zhang, Xiaoying Gao, Minh Duc Cao

Technical Report CS-TR-06/9
March 2006

Abstract

This paper describes an approach to the use of neural networks for improving the scientific paper classification performance. On the basis of the initial classification results obtained from the content-based Naive Bayes method, this approach uses neural networks to model the citation link structures of the scientific papers for refining the class labels of the documents. The approach is examined and compared with the Naive Bayes method on a standard paper classification data set with increasing training set sizes. The results suggest that using citation link structures, neural networks can significantly improve the system performance over the content-based naive Bayes method for all the training set sizes.

Keywords Document classification, content based classification, citation links

Author Information

All authors are academic staff members and postgraduate students in computer science in the School of Mathematics, Statistics and Computer Science, Victoria University of Wellington, New Zealand.

1 Introduction

As more and more scientific papers are published online these days, scientific paper classification has been becoming an important research area. It is useful for researchers, librarians, publishers to search and organise papers into necessary categories according to their need.

Several scientific paper search engines such as Google Scholar and CiteSeer provide very good tools for researchers searching scientific papers. However, searching papers on those engines is primarily based on keyword match and therefore, a search would often result in a large number of hits, majority of which may not be relevant to what users look for. A suggested solution to improve the search results is to categorise the search results into predefined topics. Users can then select the topics they are interested in. This mechanism can significantly reduce the search time.

Many machine learning approaches have been applied to general document classification. The most common methods are naive Bayes [6], decision trees [7], support vector machines [5], and neural networks [12]. However, these methods usually extract features from document *contents* only.

Scientific papers, different from general documents, do not exist in isolation but are linked together by a citation network. A paper normally cites other related published papers which are likely to have similar topics. Hence, in addition to the information from documents' own contents, the citation structure provides another source of clue that could be exploited for a better classification. We applied a simple approach using the class information of the neighbouring documents to update the document class labels obtained from the contents and the initial results were encouraging [2].

1.1 Related Work

Link analysis has been researched intensively since the birth of the WWW. Brin and Page exploited hypertext mining for PageRank, the technology behind the success of Google [1]. Oh et al.[9] combined words from connected documents into the feature vector of a document for classification. Getoor et al.[4] applied the relational model [14] for link mining and Craven et al. [3] proposed combining statistical text analysis with a relational learner such as FOIL [10]. Most of these approaches suggest that the naive use of text from neighbouring documents degrades performance. Their explanation is that link information is noisy and the distribution of terms from neighbouring documents is not sufficiently similar to the distribution of the "true" class.

1.2 Goals

This paper aims to investigate an effective approach to the use of citation link structures for further improving scientific document classification performance. In this approach, we will use the Naive Bayes method and the features extracted from document contents to determine the initial class categories of the documents, then use the citation link structures to refine the class labels. We will use neural networks for this refinement. This approach will be examined and compared with the content-based Naive Bayes classifier on a standard data set with a sequence of training sets. We will investigate how the citation links can be modelled by neural networks and whether this approach can improve the system performance of the content-based naive Bayes classifier.

1.3 Organisation

The rest of the paper first introduces the content based approach in section 2, then describes the neural network approach in section 3. After presenting the results in section 4, we conclude the paper in section 5.

2 Content Based Document Classification

The problem of document classification can be stated formally as follows. Given a corpus of n documents $\mathcal{D} = \{d_1, d_2, ..., d_n\}$ and a set of predefined m categories or classes $C = \{c_1, c_2, ...c_m\}$, assign each document d_j to one of the classes c_i.

In the content based document classification systems, each document is converted to a numeric feature vector $d_i = (w_1^i, w_2^i, ..., w_l^i)$, where l is a set of features and w_j^i is the weight of feature j for document i. The most common document representation is the *bag of words* model, where each unique word used in the corpus corresponds to a feature and the weight w_j^i reflects the number of occurrences of word j in document i.

A classifier is a function $\phi(d_j) = (s_{j1}, s_{j1}, .., s_{jm})$ and s_{ji} is the category score of document d_j assigning to class c_i. The document d_j will be assigned to the category which has the highest score:

$$\mathcal{C}(d_j) = arg\ max_i\{s_{ji}\} \tag{1}$$

In this approach, we used the naive Bayes method to train the classifier. The naive Bayes approach [6] applies the Bayes theorem to estimate the probability of a class given a test document based on the assumption that words in the documents are independent given the category. The method learns the parameters of a probabilistic model from the training data.

The classifier is trained by a set of examples S. A good training set should contain sufficient examples from all categories, so that the characteristics of all categories could be extracted by the classifier trainer.

3 Neural Networks for Document Classification Refinement

Artificial neural network is a machine learning model based on the function of biological neurons. A neural network consists of a number of artificial neurons (nodes) which are highly interconnected to each others. Each connection is associated with a weight. The neural network is trained on a number of training patterns by adjusting the weights such that the error rate is minimised.

In this approach, the Naive Bayes method uses the *contents* of the documents themselves as features to predict the initial class labels of the documents. Citation links are then used and trained by neural networks to refine the initial class labels (the results) obtained by the naive Bayes method. In the rest of this section, we describe the design of the network architecture and the network training and testing process.

3.1 Network Architecture

Multilayer feed forward neural networks have been proved to be suitable for classification and prediction problems [11, 15]. In this approach, we use a three layer network (with a single hidden layer) to perform the paper classification problem. The task then becomes determining the number of input nodes, the number of output nodes and the number of hidden nodes.

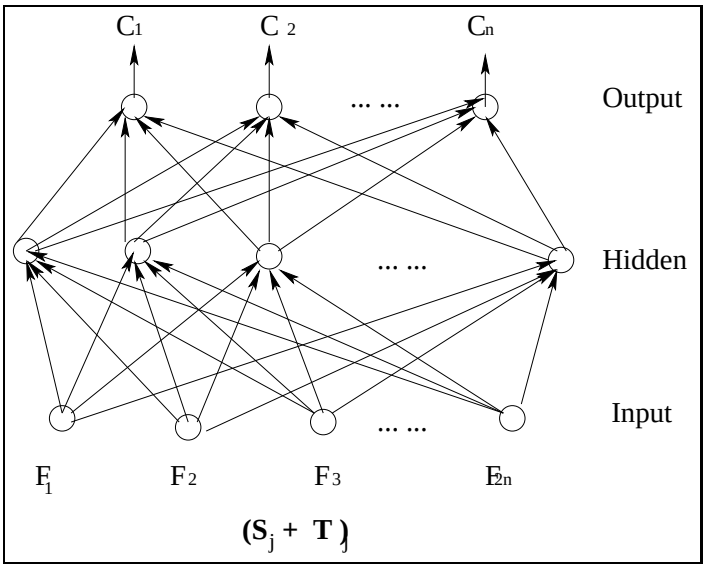

Figure 1: An example neural network.

In this approach, we use two subsets of features of a document as inputs to the neural network. The first subset $\mathbf{S}_j$ is the class label information of the document predicted by the Naive Bayes classifier, and the second subset $\mathbf{T}_j$ is citation link information of the document. The class label of each document d_j predicted by the Naive Bayes classifier is a vector $\mathbf{S}_j = (s_{j1}, s_{j2}, ..., s_{jm})$, where s_{ji} is the likelihood of document d_j for class c_i, which is used as the first subset of features.

To capture the citation link information, we use the combination of all the class labels from all the neighbouring documents obtained from the Naive Bayes classifier as the second subset of the features $\mathbf{T}_j$. The value of each feature is calculated by the sum of the scores (likelihood) of the corresponding class from all the neighbouring documents, as shown in equations 2 and 3.

$$\mathbf{T}_j = (t_{j1}, t_{j2}, ..., t_{jm}) \tag{2}$$

$$t_{jc} = \sum_{l=1}^{k} s_{lc} \tag{3}$$

where d_j is the current document, k is the number of documents that connected with document d_j and d_l is the document connected to d_j, and c is the class index. In this way, the number of features in this subset ($\mathbf{T}_j$) is the same as that of the first subset ($\mathbf{S}_j$), that is, the number of categories.

Accordingly, if the number of classes is n, then the number of output nodes will be n; the input vector size, or the total number of features that are used as inputs to neural networks, will be $2n$. In this approach, we use a single hidden layer in the network architecture. The number of nodes in the hidden layer is a trade-off between generalisation and training accuracy. In this work, we use a heuristic size of hidden layer which is three times the size of output layer, or $3n$.

3.2 Network Training and Testing

We used the back error propagation algorithm [13] with the following two variations to train the network:

- Online learning: Rather than updating the weights after presenting all the examples in a full epoch, we update the weights after presenting each training document.
- Fan-in: Weight initialisation and weight changes are modified by the *fan-in* factor. The weights are divided by the number of inputs of a node (referred to as the *fan-in* factor of the node) before network training and the value of the weight change of a node is updated accordingly during network training.

The logistic function (sigmoid function) is used as the transfer function. The sum squared error from all training patterns is used as cost function. The training process terminates when the number of training epochs reaches a predetermined number or the error rate becomes smaller than a given threshold.

During network training and testing, the network classification is considered correct if the largest activation value produced by the neural network is for the output node which corresponds to the target class. Otherwise, the classification is incorrect. For example, if the actual activation values of all the output nodes for a given document pattern is `(0.32, 0.12, 0.45, 0.85, 0.23, 0.33, 0.45)` and the target output pattern is (`0 0 0 1 0 0 0`), then this document was correctly classified as category 4 by the network; if the target output pattern is (`0 0 0 0 0 0 1`), then this document, which is one in category 7, was incorrectly classified as category 4 by the network.

4 Experiments and Results

4.1 Experiment Configuration

We used a subset of Cora, a real world scientific paper corpus [8] as the test bed in the experiments. The test bed contains 3098 papers in seven subjects of machine learning: *case based, probabilistic methods, learning theory, genetic algorithms, reinforcement learning, neural networks*, and *rule learning*. The proportion of papers in each topic ranges from 7% in the rule learning topic to 32% in the neural network topic. These papers are connected by a network of 11713 citation links. 76% of citations are between papers of same class while 24% are cross topics.

For each experiment, we randomly select an equal portion of the papers from each category to train a classifier, and use the rest as the test set to evaluate the classifier. We carry out experiments on varying training set sizes, which are 20%, 30%, 40%, 50% and 60% of the collection. For the ease of notation, we name each configuration of the data set as CORAXX where XX is the percentage of training set. For example, the set CORA40 refers to the use of 40% of the collection for training and 60% for testing.

The training set is first used to train the naive Bayes content-based classifier. The training set is then used to learn the neural network. For testing, we first apply the content based naive Bayes classifier to compute the initial class labels of the documents, then use neural networks to refine the classification results.

Each experiment is run 10 times. Note that the training set size for the 10 runs is the same but the actual documents in the training set in the 10 runs are different. For example, for each run using the set CORA20, the system randomly selects 20% of the documents in each class to make up the training set. The average results of the 10 runs on the *test sets* are presented in the next sub section.

4.2 Results

Figure 2 shows the average classification accuracy on the test sets over 10 runs of the naive Bayes method and the neural network method using different sizes of the training set.

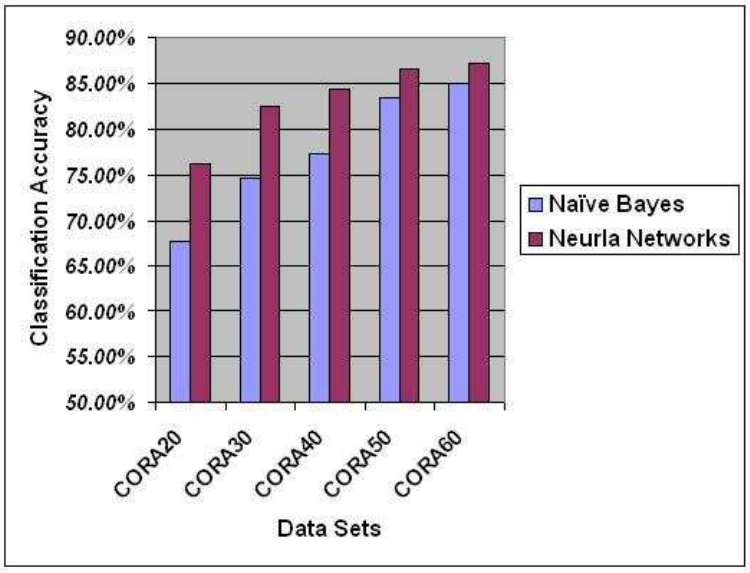

Figure 2: Results of the two methods.

As can be seen from figure 2, the neural network approach achieved significantly better classification results than the basic content based Naive Bayes method for all the data sets, suggesting that the citation link information is important for scientific paper classification. It is particularly important to use the citation based information when there are only a small number of training examples, while the content based Naive Bayes method cannot achieve satisfactory results.

As expected, for both the methods, as the training set size increased, the system performance also got improved.

5 Conclusions

The goal of this paper was to investigate a new approach to the use of the citation links to refine the initial results obtained from the content-based classifier. The goal was successfully achieved by developing a neural network based method to refine the class labels of the documents obtained by the content-based naive Bayes classification method. The new method was examined and compared with the content based Naive Bayes method on a standard paper classification data set with increasing training set sizes. The results suggest that the new approach can significantly improve the system performance and that the citation link information is important for scientific paper classification.

While the neural network method has achieved good results, we believe that it can be further improved by improving the current architecture and the current set of features. We will also investigate other approaches such as hidden Markov models and Bayesian belief networks for modelling the citation structure in the future.

Acknowledgement

We would like to thank Prof Yuejin Ma at Artificial Research Centre and College of Mechanical and Electrical Engineering, Agricultural University of Hebei, China for his providing work environment support and a number of useful discussions.

References

[1] S. Brin and L. Page. The anatomy of a Large-scale Hypertextual Web search Engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[2] M. D. Cao and X. Gao. Combining contents and citations for scientific document classification. In *Proceedings of 18th Australian Joint Conference on Artificial Intelligence*. pages 143–152, Sydney, Australia, 2005. Springer.

[3] M. Craven and S. Slattery. Relational Learning with Statistical Predicate Invention: Better Models for Hypertext. *Mach. Learn.*, 43(1-2):97–119, 2001.

[4] L. Getoor, E. Segal, B. Taskar, and D. Koller. Probabilistic Models of Text and Link Structure for Hypertext Classification. In IJCAI Workshop on Text Learning: Beyond Supervision, 2001.

[5] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*, pages 137–142, Chemnitz, DE, 1998. Springer-Verlag. LNCS, vol. 1398.

[6] D. D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. In *Proceedings of the 10th European Conference on Machine Learning*, pages 4–15, Chemnitz, DE, 1998. Springer-Verlag. LNCS, Vol.1398.

[7] D. D. Lewis and M. Ringuette. A comparison of two learning algorithms for text categorization. In *Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 81–93, Las Vegas, US, 1994.

[8] A. K. McCallum, K. Nigam, J. Rennie, and K. Seymore. Automating the Construction of Internet Portals with Machine Learning. *Information Retrieval*, 3(2):127–163, 2000.

[9] H.-J. Oh, S. H. Myaeng, and M.-H. Lee. A practical hypertext categorization method using links and incrementally available class information. In *Proceedings of the 23rd ACM International Conference on Research and Development in Information Retrieval*, pages 264–271, Athens, GR, 2000. ACM Press, New York, US.

[10] J. R. Quinlan. Learning logical definitions from relations. *Mach. Learn.*, 5(3):239–266, 1990.

[11] M. W. Roth. Survey of neural network technology for automatic target recognition. *IEEE Transactions on neural networks*, 1(1):28–43, March 1990.

[12] M. E. Ruiz and P. Srinivasan. Hierarchical neural networks for text categorization. In *Proceedings of the 22nd ACM International Conference on Research and Development in Information Retrieval*, pages 281–282, Berkeley, US, 1999. ACM Press, New York, US.

[13] D. E. Rumelhart, J. L. McClelland, and the PDP research group, editors, *Parallel distributed Processing, Explorations in the Microstructure of Cognition, Volume 1: Foundations*, chapter 8. The MIT Press, 1986.

[14] B. Taskar, E. Segal, and D. Koller. Probabilistic Classification and Clustering in Relational Data. In B. Nebel, editor, *Proceeding of IJCAI-01, 17th International Joint Conference on Artificial Intelligence*, pages 870–878, Seattle, US, 2001.

[15] Y. Won, P. D. Gader, and P. C. Coffield. Morphological shared-weight networks with applications to automatic target recognition. *IEEE Transactions on neural networks*, 8(5):1195–1203, September 1997.