Proceedings of the Fifth International Conference on Machine Learning and Cybernetics, Dalian, 13-16 August 2006

AN OPTIMAL SVM-BASED TEXT CLASSIFICATION ALGORITHM

ZI-QIANG WANG, XIA SUN, DE-XIAN ZHANG, XIN LI

School of Information and Engineering, Henan University of Technology, ZhengZhou 450052, China
E-MAIL: wzqagent@xinhuanet.com , wzqagent@126.com

Abstract:

The goal of a text classification system is to determine whether a given document belongs to which of the predefined categories. An optimal SVM algorithm for text classification via multiple optimal strategies is proposed in this paper. The experimental results indicate that the proposed optimal classification algorithm yields much better performance than other conventional algorithms.

Keywords:

Text classification; SVM; optimal strategies

1. Introduction

The amount of online document has grown greatly in recent years because of the increase in popularity of the World Wide Web. As a result, it is great importance to develop methods for the automatic processing of large collections of text documents. Text categorization (TC) is an important tool for organizing documents into classes by applying statistical methods or artificial intelligence techniques. By this, the utilization of the documents can be expected to be more effective. As a result, the situation of information overload may be alleviated. The aim of document categorization is to assign a number of appropriate categories into a textual document based on the content. This categorization process has many applications such as document routing, dissemination, or filtering [1]. A large number of techniques have been developed for text classification, including Naive Bayes [2], Nearest Neighbor [3], neural networks [4], regression [5], rule induction [6], and Support Vector Machines (SVM)[7,8]. Among them SVM has been recognized as one of the most effective text classification methods. Yang & Liu [9] provides a comprehensive comparison of supervised machine learning methods for text classification.

Due to the exponential growth of documents on the Internet and the emergent need to organize them, the automated categorization of documents into predefined labels has received an ever-increased attention in the recent years. Formally, TC consists of determining whether a document d_i (from a set of documents *D*) belongs or not to a category consistently with the knowledge of the correct categories for a set of training documents [10]. To perform this task, two main processes need to be carried out. First, the documents must be transformed into a form suitable for automatic processing. This chore includes the removal of tags, the elimination of non-informative words. Then, a measure of the importance of each stem in the document, or in most relevant words are selected according to it and used to represent the documents. This reduction of the vocabulary considered to represent the documents is a key step in the process, since it has been noted that it can greatly improve the overall performance. Once the representation of the documents is fixed, the second task is the assignment of the documents to the categories. Usually, an automatic classifier is induced from a set of correctly labeled examples using statistical methods or machine learning algorithms. Then, this classifier is used to assign each new document to one or more categories.

The goal of a text classification system is to determine whether a given document belongs to any of the predefined categories. Since the document can belong to zero, one, or more categories, the system can be a collection of binary classifiers, in which one classifier classifies for one category. One of the latest approaches and the one showing the best results so far is support vector machine (SVM)[7]. These are universal binary classifiers able to find linear or non-linear threshold functions to separate the examples of a certain category from the rest, and are based on the Structural Minimization Risk principle from computational learning theory. It has been suggested that the main reason for the good performance of SVM in TC might be the possibility of managing large spaces of features and high generalization ability. In this report, we explore this hypothesis.

2. Document representation

All the algorithms applied in TC need the documents to be represented in a way suitable for the inducing of the

1-4244-0060-0/06/$20.00 ©2006 IEEE

classifier. For SVMs, inverse document frequency(Idf) is the most commonly used importance weight. Idf has the advantage of being easy to calculate. Its disadvantage is that it disregards the frequencies of the terms within each document. For example, a term which occurs once in all documents except one, but occurs a hundred times in the one remaining document, should be judged as important for the classification of documents, because it is highly specific to one document. Therefore, we consider the empirical distribution of a type over the documents in the collection and define the importance weight w_k of term t_k by

$$w_k = \log N + \sum_{i=1}^{N} \frac{f(t_k, d_i)}{f(t_k)} \log \frac{f(t_k, d_i)}{f(t_k)} \quad (1)$$

where $f(t_k, d_i)$ is the frequency of occurrence of term t_k in document d_i, $f(t_k)$ is the frequency of occurrence of term t_k in the document collection, and *N* is the number of documents in the collection. This yields a vector of importance weights for the whole document collection:

$$w = (w_1, w_2, \cdots, w_n) \quad (2)$$

The advantage of this representation over inverse document frequency is that it does not simply count the documents a term occurs in but takes into account the frequencies of occurrence in each of the document.

3. Feature selection

The original feature space transformed with the vector space model may contain tens of thousands of different features, and not all classifiers can handle such a high dimension gracefully. Feature selection is usually employed to reduce the size of the feature space to an acceptable level, typically several orders of magnitude smaller than the original one.

Given a set of candidate terms,we select features from the set using the entropy weighting scheme[12].The entropy weighting scheme on each term is calculated as $L_{jk} \times G_j$, where L_{jk} is the local weighting of term *k* and G_j is the global weighting of term *k*. The definitions of L_{jk} and G_j are given by:

$$L_{jk} = \begin{cases} 1 + \log TF_{jk} & : TF_{jk} > 0 \\ 0 : & TF_{jk} = 0 \end{cases} \quad (3)$$

$$G_j = \frac{1 + \sum_{j=1}^{N} \frac{TF_{jk}}{F_j} \log \frac{TF_{jk}}{F_j}}{\log N} \quad (4)$$

where *N* is the number of document in a collection and TF_{jk} is the term frequency of each word in the document Doc_j .The F_j is a frequency of term *j* in the entire document collection. We choose terms with the highest entropy value to be an input to the classifier for classification.

After the feature selection process, the trained classifier is ready to be used for categorizing a new document. For the classifiers in the text classification algorithm, we adopt support vector machines that show significant improvement over other machine learning algorithms when applied to document classification.

4. Text classification with SVM

Support vector machine (SVM) is a powerful supervised learning paradigm based on the structured risk minimization principle from computational learning theory. Which is a state of the art classification algorithm that is known to be successful in a wide variety of applications. High generalization ability of the method makes it particularly suited for high dimensional data such as text. Indeed, it has been shown that SVM outperformed most of the other classification algorithms in text categorization tasks.

Classification algorithm: Suppose we are given a set of examples $(x_1, y_1), \cdots, (x_l, y_l), -x \in R^N, y_i \in \{-1, +1\}$. We consider decision functions of the form $\mathrm{sgn}((w \cdot x) + b)$, where $(w \cdot x)$ represents the inner product of *w* and *x*. We would like to find a decision function $f_{w,b}$ with the properties

$$y_i((w \cdot x) + b) \geq 1, \quad i = 1, \cdots, l \quad (5)$$

In many practical situations, a separating hyperplane does not exist. To allow for possibilities of violating (5), slack variables are introduced like

$$\xi_i \geq 0, \quad i = 1, \cdots, l \quad (6)$$

to get

$$y_i((w \cdot x) + b) \geq 1 - \xi_i, \quad i = 1, \cdots, l \quad (7)$$

The support vector approach for minimizing the generalization error consists of the following:

$$\text{Minimize: } \phi(w, \xi) = (w \cdot w) + C \sum_{i=1}^{l} \xi_i \quad (8)$$

subject to constraints (6) and (7).

It can be shown that minimizing the first term in (8) amounts to minimizing a bound on the VC-dimension and minimizing the second term corresponds to minimizing the misclassification error. The above minimization problem

can be posed as a constrained quadratic programming (QP) problem, which can be formulated as the following convex quadratic programming problem:

$$\text{Maximize: } \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j x_i^T x_j \qquad (9)$$

subject to

$$0 \le \alpha_i \le C(i=1,\cdots,m), \sum_{i=1}^{l}\alpha_i y_i = 0 \qquad (10)$$

where α_i are Lagrange multipliers, C is a parameter that assigns penalty cost to misclassification of samples. By solving the above optimization problem, the solution gives rise to a decision function of the form:

$$f(x) = \text{sgn}\left[\sum_{i=1}^{l} y_i\alpha_i(x \cdot x_i) + b\right] \qquad (11)$$

where b is a bias term. Only a small fraction of the coefficients α_i are nonzero. The corresponding pairs of entries x_i are known as support vectors and they fully define the decision function.

5. Performance comparison

To evaluate the effectiveness of our approach, comprehensive performance study has been conducted using two standard text dataset: Reuter-21578[13] and TREC-AP.

The Reuters-21578 test collection contains documents collected from the Reuters newswire in 1987. It is a standard text categorization benchmark and contains 135 categories. However, we only examined the ten most frequent classes since small numbers of testing examples makes estimating some performance measures unreliable due to high variance. For the experiments below we used only the top 300 words with highest mutual information for each class; approximately 15K words appear in at least three training documents.

The TREC-AP corpus is a collection of AP news stories from 1988 to 1990. We used the same train/test split of 142791/66992 documents that was used by Ref.[14] . For the experiments described below, we use only the top 1000 words with the highest mutual information for each class; approximately 123K words appear in at least 3 training documents.

In addition to varying the number of features, SVM was used as the classification algorithm, its performance is governed by two parameters, C (the penalty imposed on training examples that fall on the wrong side of the decision boundary), and p (the decision threshold).

To improve the classification performance, we adopt the following method to tune the value of C. Each data set was split into 70% and 30% for training and testing respectively. This was done 15 times using a stratified sampling scheme to obtain 15 different trials for each dataset. Those parameters of the SVM were tuned in steps of 2^k, for the parameter C, k goes from 1 to 11. A 3-fold cross validation was carried out to determine the training set accuracy at the various parameter settings. The parameter that gave the best training accuracies was then used to determine the accuracy values for each of the corresponding test sets.

The decision threshold, p, can be set to control precision and recall for different tasks. Increasing p, results in fewer test items meeting the criterion, and this usually increases precision but decreases recall. Conversely, decreasing p typically decreases precision but increases recall. We choose $p = 0.5$ so as to optimize performance on the F1 measure on a training validation set. The SVM classifier was implemented using the LIBSVM software.

To compare the performance of the classification methods, we look at a set of standard performance measures. We use the F1 measure introduced by Ref.[1], this measure is the harmonic mean of precision and recall, which combines recall and precision in the following way:

$$\text{Re}\,call = \frac{\text{number of correct positive prediction}}{\text{number of positive examples}}$$

$$\text{Pr}\,ecision = \frac{\text{number of correct positive prediction}}{\text{number of positive prediction}}$$

$$F1 = \frac{2\times\text{Recall}\times\text{Precision}}{(\text{Recall}+\text{Precision})}$$

For F1, we can use either macro-average or micro-average. In macro-averaging, the score is computed separately for each class and then arithmetically averaged; this tends to weight rare classes more heavily. Micro-averaged values are computed directly from the binary decisions over all classes; this places more weight on the common classes. We evaluated the systems with both macro and micro averaged F1.

In addition, we computed and displayed a receiver-operating characteristic (ROC) curve, which represents the performance of a classifier under any linear utility function [15]. We report results on the area under the ROC curve as an attempt to summarize the linear utility space of functions.

We split the each dataset into three parts. Then we use two parts for training and the remaining third for test. We conduct the training-test procedure three times and use the average of the three performances as final result. This is so called three-fold cross validation.

In this evaluation, we compared our proposed optimal SVM(OSVM) method with decision-tree algorithm (DTA) [6], naive Bayes classifier(NBC)[2] and K-nearest neighbor algorithm(KNN)[9] on Reuter-21578 and TREC-AP. In Table 1 and Table 2, we present the main performance results over the two corpora. In terms of the various performance measures, better performance is indicated by larger F1 or ROC area values. From the two tables we can see that our proposed method outperforms other conventional classification algorithm.

Table 1. Performance on Reuters corpus

Method	Macro-F1	Micro-F1	ROC
DTA	0.7847	0.8541	0.9803
NBC	0.6575	0.7906	0.9705
KNN	0.6738	0.7815	0.9658
OSVM	0.8492	0.9124	0.9831

Table 2. Performance on TREC-AP

Method	Macro-F1	Micro-F1	ROC
DTA	0.6014	0.5796	0.9763
NBC	0.5675	0.5351	0.9754
KNN	0.6013	0.5427	0.9697
OSVM	0.7348	0.6968	0.9845

6. Conclusions

In this paper we develop an optimal SVM algorithm via multiple optimal strategies, such as a novel importance weight definition, the feature selection using the entropy weighting scheme, the optimal parameter settings, etc. Comparison between our method and other conventional text classification algorithms is conducted on Reuter and TREC corpora. The experimental results indicate that our proposed algorithm yields much better performance than other conventional algorithms. Consequently, our proposed algorithm is an effective algorithm for text classification problems.

Acknowledgements

This paper is supported by the National Natural Science Foundation of China under Grant No.90412013-3, the Natural Science Foundation of Henan Province under Grant No.0511011700, and the Natural Science Foundation of Henan Province Education Department under Grant No. 200510463007.

References

[1] C.J. van Rijsbergen, Information Retrieval, Butterworths, London, 1979.

[2] D.D.Lewis, Naive bayes at forty: the independence assumption in information retrieval, Proceedings of the 10th European Conference on Machine Learning, Chemnitz, pp.4-15,April 1998.

[3] Y.Yang, An evaluation of statistical approaches to text categorization, Information Retrieval, Vol.1, No.1, pp. 69-90,April 1999.

[4] T.H.Ng, W.B.Goh, and K.L.Low, Feature selection, perception learning and a usability case study for text categorization, Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Philadelphia, pp.67-73,July 1997.

[5] Y.Yang, C.G.Chute, An example-based mapping method for text categorization and retrieval, ACM Transactions on Information Systems, Vol.12, No.3, pp.252-277,July 1994.

[6] C. Apte, F.Damerau, and S.M.Weiss, Automated learning of decision rules for text categorization, ACM Transactions on Information Systems, Vol.12, No.3, pp.233-251,July 1994.

[7] V. N.Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, New York,1995.

[8] T.Joachims, Text categorization with support vector machines: learning with many relevant features, Proceedings of the 10th European Conference on Machine Learning,Chemnitz,pp.137-142,April 1998.

[9] Y.Yang,and X.Liu, An re-examination of text categorization, Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Berkeley, pp.42-49,August 1999.

[10] F. Sebastiani, Machine learning in automated text categorization, ACM Computing Surveys, Vol.34, No. 1, pp.1-47, March 2002.

[11] S.Dumais, J.Platt, D. Heckerman, and M. Sahami, Inductive learning algorithms and representations for text categorization, Proceedings of the 1998 ACM CIKM International Conference on Information and Knowledge Management, Bethesda, pp.148-155, November 1998.

[12] S.T. Dumais, Improving the retrieval of information from external sources, Behavior Research Methods, Instruments and Computers, Vo.23,No.2,pp.229-236, February 1991.

[13] D.D.Lewis,Reuters-21578, distribution 1.0, http: // www. daviddlewis. com/resources/testcollections/ reuters 21578,1997.

[14] D.D.Lewis, R.E.Schapire, J.P. Callan, and R. Papka, Training algorithms for linear text classifiers, Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Zurich, pp.298-306, August 1996.

[15] F.J.Provost, T.Fawcett, Robust classification for imprecise environments, Machine Learning, Vol.42, No.3, pp.203-231,March 2001.