行政院國家科學委員會專題研究計畫成果報告

支援向量法解模型辨識問題

Support Vector Machines for Pattern Recognition

計畫編號：NSC 89-2213-E-002-013
執行期限：88 年 8 月 1 日至 89 年 7 月 31 日
主持人：林智仁　　台灣大學資訊工程系
e-mail: cjlin@csie.ntu.edu.tw
http://www.csie.ntu.edu.tw/~cjlin

一、中文摘要

支援向量法(Support vector machines)是一個非常新而且有前景的模型辯識方法。 經過 Bell Labs 人員在 90 年代初期的研究，目前 Support vector machines (SVM) 已成為 Machine Learning 與 Pattern Recognition 中一重要主題

解 SVM 的最重要步驟是解一個大型的二次最佳化問題。因 storage 的限制，傳統的數學歸劃方法在此完全無法使用。目前少數能解 SVM 的方法之一為 decomposition method。從 97 年被提出後，其收斂性一直是個未解決的問題。在本計畫中，我們提出了收斂性的證明。我們的方法甚已正被其他的研究人員推廣至 SVM regression。在有了理論的經驗之後，我們也做了一系列數值上的分析。我們已 release 兩套 SVM 軟體供研究人員及一般工業界使用。

關鍵詞：支援向量法,模型辨識,統計學習,最佳化問題

Abstract

The support vector machine (SVM) is a new and very promising technique for classification and pattern recognition. After the development at Bell Labs in early 90s, it has now been an important research field in machine learning and pattern recognition.

The success of support vector machines depends on solving a large dense quadratic programming problem. The memory restriction of storing dense matrices makes traditional optimization algorithms useless. Currently the major method to solve this special problem is the decomposition method. Since its introduction in 1997, the convergence had been an open issue. In this project we solved this difficult problem. In In addition, after extensive numerical experiments, we proposed new methods and two software which are available to researchers/engineers worldwide.

Keywords: pattern recognition, support vector machines, statistical learning theory, large-scale optimization

二、Introduction

The support vector machine (SVM) [1,7] is a new and promising technique for pattern recognition. It has been developed by Vapnik and his group at former AT&T Bell Laboratories in early 90s. SVM is an approximate implementation of the structural risk inimization (SRM) induction principle by Vapnik. Therefore, this topic belongs to the field of statistical learning theory. Some applications are, for example, object recognition, combustion engine detection, protein sequences, function estimation, text categorization, chaotic system, handwritten digit recognition, and database marketing.

Given l training data in two classes, the support vector machine tries to find a separating hyperplane between these two classes of data. Though this idea for linear separable data has been proposed 30 years ago, until recently researchers realized how to generalize it to more complicated data. To be more precise, for data which are not linear separable, we can map them to a higher dimensional space. Then on this higher space, there may have a linear separating hyperplane to separate those data.

To find the separating hyperplane, the support vector machine requires the solution of a quadratic programming problem with l variables. However, the matrix in the objective function is fully dense so there is even a problem of storing the problem itself. Therefore, traditional optimization methods cannot be applied to solve this problem.

To solve large-scale classification problems, after 1997, some researchers have proposed methods to conquer this memory difficulty. A major method is called the decomposition method where the basic idea is to fix most variables and only work on few variables in each optimization iteration. That is, a sub-problem is solved so the memory is enough to store the data of this sub-problem. This is like that if you want to minimize a function with ten variables, you use an iterative process where in each iteration, eight variables are fixed and a sub-problem with two variables is solved. Thus we call these two variables the working set while the other eight variables are in the non-working sets. In each iteration, there will have a new working set. Then each variable is updated in some iterations so finally the problem is minimized.

After the decomposition method was proposed, there are several successful implementations. However, until we gave the proof, the convergence of the decomposition method was an open issue. That is, without a serious proof, we do not know if the iterative process will converge to an optimal solution or converge/diverge to somewhere else.

Our proof uses an optimization technique called ``projected gradient.'' It has appeared in the following paper [3]. Many people think this is an important work as it shows the validity of the decomposition method.

Some continuing research on the convergence issue has been conducted by other researchers (e.g. [5]) and ourselves [6].

After the theoretical issues are handled, we are interested in practical software. We found out that for existing decomposition method, when using some parameters, the convergence is very slow. For a problem with hundreds of data, the decomposition method may take tens of thousands iterations. We think a better working set can improve this situation. In the paper, we did some extensively experiments and analysis on the numerical properties of the decomposition method. We then proposed a new working set selection strategy [4] which leads to faster convergence for difficult cases. This working selection always tries to push variables to lower and upper bounds as much as possible. Thus the number of free variables, that is, variables strictly between lower and upper bounds, is kept as small as possible.

The new working selection was implemented in the software BSVM [4]. A detailed description of it is in the following paper: . Since its introduction in February 2000, more then 1,500 researchers have downloaded it. Some sample users are, for example, US army research laboratory, and researchers for MIT, etc.

After the release of BSVM, some researchers not in the fields of pattern recognition and machine learning complained that our software is a little too complicated for them to use. Indeed many users of support vector machines do not know much about optimization. Motivated by users' comments, we designed a simple SVM software: LIBSVM [2]. Though it may not be the most efficient in all cases, it is very simple and easy-to-use. In less than one thousand C++ code, an SVM classifier was implemented.

LIBSVM was released in April 2000. Up to October 2000, there have been 1,700 downloads. Researchers in Ohio State University even built a MATLAB interface on it.

Though several applications of support vector machines have appeared, there are still gaps on making SVM practically useful. Right now we think two bottlenecks are:

1. Multi-class classification: Originally support vector machines were designed for two-class problems. Hence we have to separate a multi-class problem to several two-class problems before using support vector machines. Up to now, there are still no effective implementations for multi-class data.
2. Automatic model selection: The support vector machine needs several parameters where users would like to choose a parameter set which gives the best generalization performance. As training vectors are data with known classes, it is not that important to minimize the training error. Instead, the average error rates of using test data (that is, generalized error) count. The generalization error is difficult to know so is usually approximated by some estimators. However, these estimators need excessive computational time so up to now there are no support vector machine software which can do effective automatic model selections.

We have started to investigate these two issues. Some temporary results are very promising. We hope that eventually the support vector machine can be as useful as tools like neural networks.

四、Conclusions

In this report we have summarized our past efforts on support vector machines. This is a very active research topic so we will continue to work on it in the next NSC project.

五、Acknowledgements

Students who join this projects are Chih-Chung Chang, Chih-Wei Hsu, and Jen-Hou Lee.

六、Bibliography

[1] C. J. C. Burges "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery,* 2(1998) 121-167.

[2] C.-C. Chang and C.-J. Lin. "LIBSVM: introduction and benchmarks," (2000) Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[3] C.-C. Chang, C.-W. Hsu and C.-J. Lin. "The analysis of decomposition methods for support vector machines," *IEEE Transactions on Neural Networks.* 11(2000), 1003-1008.

[4] C.-W. Hsu and C.-J. Lin. "A simple decomposition method for support vector machines," submitted to *Machine Learning.* (2000). Software available at http://www.csie.ntu.edu.tw/~cjlin/bsvm

[5] S. Keerthi and E. G. Gilbert. "Convergence of a generalized SMO algorithm for SVM classifier design," Technical Report CD-00-1, Department of Mechanical and Production Engineering, National University of Singapore, 2000.

[6] C.-J. Lin. "On the convergence of the decomposition method for support vector machines," Technical Report, Department of Computer Science and Information Engineering, National Taiwan University, 2000. Submitted to IEEE Transactions on Neural Networks.

[7] V. Vapnik. Statistical Learning Theory. John Wiley, New York, 1998.