BIOINFORMATICS APPLICATIONS NOTE

Vol. 20 no. 4 2004, pages 586–587
DOI: 10.1093/bioinformatics/btg461

Support vector machine classification on the web

Paul Pavlidis[1,*], Ilan Wapinski[2,†] and William Stafford Noble[3]

[1]Columbia Genome Center and Department of Biomedical Informatics, Columbia University, 1150 St Nicholas Avenue, New York, NY 10032, USA, [2]Department of Computer Science, Columbia University, New York, NY 10027, USA and [3]Department of Genome Sciences, University of Washington, 1705 NE Pacific Street, Seattle, WA 98195, USA

Received on August 13, 2003; revised on October 3, 2003; accepted on October 6, 2003
Advance Access publication January 22, 2004

ABSTRACT

Summary: The support vector machine (SVM) learning algorithm has been widely applied in bioinformatics. We have developed a simple web interface to our implementation of the SVM algorithm, called Gist. This interface allows novice or occasional users to apply a sophisticated machine learning algorithm easily to their data. More advanced users can download the software and source code for local installation. The availability of these tools will permit more widespread application of this powerful learning algorithm in bioinformatics.
Availability: Web interface at svm.sdsc.edu. Binaries and source code at microarray.cpmc.columbia.edu/gist.
Contact: pp175@columbia.edu

The support vector machine (SVM) (Vapnik, 1998) is a supervised learning algorithm, useful for recognizing subtle patterns in complex datasets. The algorithm performs discriminative classification, learning by example to predict the classifications of previously unseen data. The algorithm has been applied in domains as diverse as text categorization, image recognition and hand-written digit recognition (Cristianini and Shawe-Taylor, 2000). Recently, SVMs have been applied in numerous bioinformatics domains [reviewed by Noble (2004)], including recognition of translation start sites (Zien *et al.*, 2000), protein remote homology detection (Jaakkola *et al.*, 1999; Liao and Noble, 2002; Leslie *et al.*, 2003), protein fold recognition (Ding and Dubchak, 2001), microarray gene expression analysis (Brown *et al.*, 2000; Guyon *et al.*, 2001; Mukherjee *et al.*, 1999; Furey *et al.*, 2001; Vert and Kanehisa, 2003), functional classification of promoter regions (Pavlidis *et al.*, 2001), prediction of protein–protein interactions (Bock and Gough, 2001) and peptide identification from mass spectrometry data (Anderson *et al.*, 2003).

The popularity of the SVM algorithm stems from four primary factors. First, the algorithm boasts a strong theoretical foundation, based upon the dual ideas of VC dimension and structural risk minimization (Vapnik, 1998). Second, the SVM algorithm scales well to relatively large datasets. Third, the SVM algorithm is flexible, as evidenced by the list of applications above. This flexibility is due in part to the robustness of the algorithm itself, and in part to the parameterization of the SVM via a broad class of functions, called kernel functions. The behavior of the SVM can be modified to incorporate prior knowledge of a classification task simply by modifying the underlying kernel function. The fourth and most important explanation for the popularity of the SVM algorithm is its accuracy. Although the underlying theory suggests explanations for the SVMs excellent learning performance, its widespread application is due in large part to the empirical success the algorithm has achieved.

This note describes a publicly accessible web interface that allows novice or occasional SVM users to perform SVM training and classification. For details on using the software and explanations of the underlying algorithms, we refer readers to the web site and the references listed there. Users who find the web interface limiting can use the command-line Gist software, which is available in binary and source code form. Potential SVM users might also be interested in a number of other liberally licensed SVM implementations that have been described previously, including mySVM (http:www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVM), SVMlight (Joachims, 1998), LIBSVM (www.csie.ntu.edu.tw/~cjlin/libsvm) and svmTorch (Collobert and Bengio, 2001).

The entry point to the SVM is a simple form that has only three required inputs: a training dataset, a set of class labels for the training data and a test dataset. The datasets take the form of spreadsheet-like, tab-delimited text files that are very simple to set up. The training and test data files each consist of examples given as rows of tab-delimited features. For example, for a tumor-classification task using gene expression data, each row represents an individual tumor sample, and

[*] To whom correspondence should be addressed.

[†] Present address: Division of Engineering and Applied Sciences and Center for Genome Research, Harvard University, Cambridge, MA 02138, USA

each column represents the expression level of a particular gene. The class label file identifies each training example as either a positive (denoted by ‘1’) or negative (‘−1’). A demonstration dataset is available on the web site, and a sample analysis using this dataset can be run by checking a box on the input page.

Upon submitting their data, the user is directed to a page that informs them of the progress of the analysis and any problems encountered. When the SVM analysis is complete, the results are presented both as HTML tables and as down-loadable, tab-delimited files that can be used for further analysis by the user. The results consist of two parts: training results and test results. For training, a summary is provided in terms of total number of errors committed (false positives etc.), as well as a detailed table of the results for each example. The test results consist of a predicted classification of each test example.

The web interface provides a number of parameters that can be optionally set by the user. Among the most important is the choice of the kernel function. Gist implements the commonly used polynomial and radial basis function kernels; the default is a simple dot product. Using higher-order polynomial or radial basis kernels can sometimes improve the separability of the two classes of samples by performing an implicit projection of the data into a higher-dimensional feature space.

Gist also implements a so-called ‘soft margin’, meaning that classification errors during training can be tolerated. This makes the algorithm capable of learning from noisy datasets that might otherwise be non-separable. The default soft margin settings work well in many cases we have encountered, but the user has a choice of both one-norm and two-norm soft margins (Cristianini and Shawe-Taylor, 2000). These and other parameters are fully documented on the web site. Additional features of the software not available in the web interface, including hold-one-out cross-validation and feature selection, can be accessed by using the command-line version of Gist.

ACKNOWLEDGEMENTS

We thank Andrew Liu and Phan Lu for contributions to an early version of the web interface. This work is supported by a Sloan Research Fellowship to W.S.N., and by National Science Foundation grants DBI-0078523 and ISI-0093302. We are grateful for computational resources provided by the National Biomedical Computation Resource at the San Diego Supercomputer Center. NBCR is funded by the NIH National Center for Research Resources (P41 RR08605).

REFERENCES

Anderson,D.C., Li,W., Payan,D.G. and Noble,W.S. (2003) A new algorithm for the evaluation of shotgun peptide sequencing in proteomics: support vector machine classification of peptide MS/MS spectra and SEQUEST scores. *J. Proteome Res.*, **2**, 137–146.

Bock,J.R. and Gough,D.A. (2001) Predicting protein–protein interactions from primary structure. *Bioinformatics*, **17**, 455–460.

Brown,M.P.S., Grundy,W.N., Lin,D., Cristianini,N., Sugnet,C., Furey,T.S., Ares,J.M. and Haussler,D (2000) Knowledge-based analysis of microarray gene expression data using support vector machines. *Proc. Natl Acad. Sci., USA*, **97**, 262–267.

Collobert,R. and Bengio,S. (2001) Svmtorch: support vector machines for large-scale regression problems. *J. Machine Learn. Res.*, **1**, 143–160.

Cristianini,N. and Shawe-Taylor,J. (2000) *An Introduction to Support Vector Machines*. Cambridge University Press, MA.

Ding,C. and Dubchak,I. (2001) Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, **17**, 349–358.

Furey,T.S., Cristianini,N., Duffy,N., Bednarski,D.W., Schummer,M. and Haussler,D. (2001) Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, **16**, 906–914.

Guyon,I., Weston,J., Barnhill,S. and Vapnik,V. (2001) Gene selection for cancer classification using support vector machines. *Machine Learning*, **46**, 389–422.

Jaakkola,T., Diekhans,M. and Haussler,D. (1999) Using the Fisher kernel method to detect remote protein homologies. *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, AAAI Press, Menlo Park, CA, pp. 149–158.

Joachims,T. (1998) Making large scale svm learning practical. *Technical Report LS8-24*, Universitat Dortmund.

Leslie,C., Eskin,E., Weston,J. and Noble,W.S. (2003) Mismatch string kernels for SVM protein classification. In Becker,S., Thrun,S. and Obermayer,K. (eds), *Advances in Neural Information Processing Systems*. MIT Press, Cambridge, MA.

Liao,L. and Noble,W.S. (2002) Combining pairwise sequence similarity and support vector machines for remote protein homology detection. *Proceedings of the Sixth Annual International Conference on Computational Molecular Biology*, pp. 225–232.

Mukherjee,S., Tamayo,P., Slonim,D., Verri,A., Golub,T., Mesirov,J. and Poggio,T. (1999) Support vector machine classification of microarray data. *Technical Report AI Memo 1677*, Massachusetts Institute of Technology.

Noble,W.S. (2004) Support vector machine applications in computational biology. *Kernel Methods in Computational Biology*, MIT Press, Cambridge, MA.

Pavlidis,P., Furey,T.S., Liberto,M., Haussler,D. and Grundy,W.N. (2001) Promoter region-based classification of genes. In Altman,R.B., Dunker,A.K., Hunter,L., Lauderdale,K., and Klein,T.E., (eds), *Pacific Symposium of Biocomputing*, Singapore, pp. 151–163. World Scientific.

Vapnik,V.N. (1998) *Statistical Learning Theory. Adaptive and Learning Systems for Signal Processing, Communications, and Control*. Wiley, New York.

Vert,J.-P. and Kanehisa,M. (2003) Graph-driven features extraction from microarray data using diffusion kernels and kernel CCA. In Becker,S., Thrun,S. and Obermayer,K. (eds), *Advances in Neural Information Processing Systems 15*. MIT Press, Cambridge, MA.

Zien,A., Rätch,G., Mika,S., Schölkopf,B., Lengauer,T. and Müller,K.-R. (2000) Engineering support vector machine kernels that recognize translation initiation sites. *Bioinformatics*, **16**, 799–807.