# SPEEDING THE TRAINING OF SUPPORT VECTOR MACHINES


**Jin Hyuk Jung**
*Computer Science Department*
*University of Maryland,*
*College Park, MD 20742*
*U.S.A.*
*e-mail:* jjung@cs.umd.edu

**Dianne P. O'Leary**
*Computer Science Department and Institute for Advanced Computer Studies,*
*University of Maryland,*
*College Park, MD 20742*
*U.S.A.*
*e-mail:* oleary@cs.umd.edu

**André L. Tits**
*Department of Electrical and Computer Engineering and*
*Institute for Systems Research*
*University of Maryland*
*College Park, MD 20742*
*email:* andre@umd.edu



**Keywords:** SVM, support vector machines, quadratic programming, clustering



### Abstract

Data classification is a fundamental task in science and engineering. For example, given data gathered about a patient's tumor, we might need to decide whether the tumor is malignant or benign. Ideally, we would like to determine a mathematical function whose evaluation would indicate the classification of the tumor. Linear discriminant analysis provides one such function, but functions more general than a separating hyperplane are needed in many applications.

Support vector machines (SVMs) provide a means to classify data into two groups (positive and negative) using criteria more descriptive than separating hyperplanes. SVMs are *trained* using a large set of positive and negative examples. Classifiers such as neural networks are trained by an iterative process of presenting examples and adjusting network weights until convergence. In contrast, the training of an SVM is acccomplished by solving a single quadratic programming problem whose size is determined by the number of examples and the number of parameters in the classifier. This simple training regime is a major advantage of the SVM framework.

These quadratic programming problems can, however, be quite large. In this work we use two approaches to improve computational efficiency. First, we apply an adaptive constraint reduction method in an interior point method for solving the quadratic programming problem. This has the effect of allowing later iterations to focus on only a few of the example datapoints. Second, we cluster the data and initially train on a small number of examples drawn from each cluster. This has the effect of reducing computation time in the early iterations.

We discuss our algorithm and its convergence theory and illustrate its performance on a variety of examples.

**Acknowledgement:** This research was supported by the US Department of Energy under grant DEFG0204ER25655.