Tuning Support Vector Machines for Minimax and Neyman-Pearson Classification

Mark A. Davenport, *Student Member*, *IEEE*, Richard G. Baraniuk, *Fellow*, *IEEE*, and Clayton D. Scott, *Member*, *IEEE*

Abstract—This paper studies the training of support vector machine (SVM) classifiers with respect to the minimax and Neyman-Pearson criteria. In principle, these criteria can be optimized in a straightforward way using a cost-sensitive SVM. In practice, however, because these criteria require especially accurate error estimation, standard techniques for tuning SVM parameters, such as cross-validation, can lead to poor classifier performance. To address this issue, we first prove that the usual cost-sensitive SVM, here called the $2C$-SVM, is equivalent to another formulation called the 2ν-SVM. We then exploit a characterization of the 2ν-SVM parameter space to develop a simple yet powerful approach to error estimation based on smoothing. In an extensive experimental study, we demonstrate that smoothing significantly improves the accuracy of cross-validation error estimates, leading to dramatic performance gains. Furthermore, we propose coordinate descent strategies that offer significant gains in computational efficiency, with little to no loss in performance.

Index Terms—Minimax classification, Neyman-Pearson classification, support vector machine, error estimation, parameter selection.

1 Introduction

In binary classification, false alarms and misses typically have different costs. Thus, a common approach to classifier design is to optimize the expected misclassification (Bayes) cost. Often, however, this approach is impractical because either the prior class probabilities or the relative cost of false alarms and misses are unknown. In such cases, two alternatives to the Bayes cost are the minimax and Neyman-Pearson (NP) criteria. In this paper, we study the training of support vector machine (SVM) classifiers with respect to these two criteria, which require no knowledge of prior class probabilities or misclassification costs. In particular, we develop a method for tuning SVM parameters based on a new strategy for error estimation. Our approach, while applicable to training SVMs for other performance measures, is primarily motivated by the minimax and NP criteria.

To set notation, let $(\mathbf{x}_i, y_i)_{i=1}^n$ denote a random sample from an unknown probability measure, where $\mathbf{x}_i \in \mathbb{R}^d$ is a *training vector* and $y_i \in \{-1, +1\}$ is the corresponding *label*. For a classifier $f : \mathbb{R}^d \rightarrow \{+1, -1\}$, let

$$P_F(f) = \Pr(f(\mathbf{x}) = +1 | y = -1) \tag{1}$$

and

$$P_M(f) = \Pr(f(\mathbf{x}) = -1 | y = +1) \tag{2}$$

denote the *false alarm* and *miss* rates of f, respectively.

When there is no reason to favor false alarms or misses, a common strategy is to select a classifier operating at the *equal error rate* or the *break-even point*, where $P_F(f) = P_M(f)$ [1], [2], [3]. Of course, many classifiers may satisfy this constraint. We seek the best possible, the *minimax* classifier, which is defined as

$$f^*_{MM} = \arg\min_f \max\{P_F(f), P_M(f)\}. \tag{3}$$

An alternative approach is the NP paradigm [1], [4], which naturally arises in settings where we can only tolerate a certain level of false alarms. In this case, we seek the lowest miss rate possible provided the false alarm rate satisfies some constraint. Specifically, given a user-specified level α, the NP-optimal classifier is defined as

$$f^*_\alpha = \arg\min_{f: P_F(f) \leq \alpha} P_M(f). \tag{4}$$

Under suitable conditions on the distribution of $(\mathbf{x}, y)$, such as the class-conditional distributions being continuous, both f^*_{MM} and f^*_α are equal to the solution of

$$\min_f \gamma P_F(f) + (1 - \gamma) P_M(f), \tag{5}$$

for appropriate values of γ [5]. This suggests that training an SVM for minimax and NP classification could, in principle, be accomplished by simply using a cost-sensitive SVM and tuning the parameter γ to achieve the desired error constraints. However, tuning parameters for minimax and NP criteria is very different from tuning parameters for a Bayesian criterion like that in (5) in one critical respect: To minimize the minimax or NP criteria, one must use estimates of $P_F(f)$ and $P_M(f)$ to determine the appropriate

- *M.A. Davenport is with the Department of Statistics, Stanford University, 390 Serra Mall, Stanford, CA 94305. E-mail: md@rice.edu.*
- *R.G. Baraniuk is with the Department of Electrical and Computer Engineering, Rice University, MS-380, 6100 Main Street, Houston, TX 77005. E-mail: richb@rice.edu.*
- *C.D. Scott is with the Department of Electrical Engineering and Computer Science, University of Michigan, 1301 Beal Avenue, Ann Arbor, MI 48105. E-mail: cscott@eecs.umich.edu.*

Manuscript received 16 Aug. 2008; revised 22 Feb. 2009; accepted 5 July 2009; published online 21 Jan. 2010.
Recommended for acceptance by B. Scholkopf.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2008-08-0531.
Digital Object Identifier no. 10.1109/TPAMI.2010.29.

0162-8828/10/$26.00 © 2010 IEEE Published by the IEEE Computer Society

γ. As a result, for minimax and NP classification, it is extremely important to have accurate estimates of $P_F(f)$ and $P_M(f)$, whereas since γ is predefined for Bayesian criteria, error estimates can be less accurate (e.g., biased) and still lead to good classifiers.

To tackle the issue of accurate error estimation in cost-sensitive SVMs, we adopt a particular formulation called the 2ν-SVM [6]. We prove that this cost-sensitive SVM is equivalent to the more common $2C$-SVM [7], [8], [9] and provide a careful characterization of its parameter space in Section 2. We then leverage this characterization to develop simple but powerful approaches to error estimation and parameter selection based on smoothing cross-validation (CV) error estimates and coordinate descent search strategies in Section 3. We conduct a detailed experimental evaluation in Sections 4 and 5 and demonstrate the superior performance of 1) our approaches to estimation relative to conventional CV and 2) our approach to minimax and NP classification relative to SVM-based approaches more commonly used in practice. Section 6 concludes with a brief discussion. Our results build on those published in [10], [11], [12]. Our software—based on the LIBSVM package [13]—is available online at www.dsp.rice.edu/software.

2 Cost-Sensitive Support Vector Machines

2.1 Review of SVMs

Conceptually, a support vector classifier is constructed in a two-step process [14]. In the first step, we transform the $\mathbf{x}_i$ via a mapping $\Phi : \mathbb{R}^d \to \mathcal{H}$, where $\mathcal{H}$ is a Hilbert space. In the second step, we find the hyperplane in $\mathcal{H}$ that maximizes the *margin*—the distance between the decision boundary and the closest training vector (from either class) to the boundary. If $\mathbf{w} \in \mathcal{H}$ and $b \in \mathbb{R}$ are the normal vector and affine shift (or *bias*) defining the max-margin hyperplane, then the support vector classifier is given by $f_{\mathbf{w},b}(\mathbf{x}) = \operatorname{sgn}(\langle \mathbf{w}, \Phi(\mathbf{x})\rangle_{\mathcal{H}} + b)$.

The max-margin hyperplane is the solution of a simple quadratic program:

$$(P) \qquad \min_{\mathbf{w},b} \ \frac{1}{2}\|\mathbf{w}\|^2$$
$$\text{s.t.} \quad y_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle_{\mathcal{H}} + b) \geq 1, \quad \text{for } i = 1, \dots, n.$$

One can show via a simple geometric argument that, for any $\mathbf{w}$ satisfying the constraints in (P), the two classes are separated by a margin of $2/\|\mathbf{w}\|$; hence, minimizing the objective function of (P) is equivalent to maximizing the margin. This problem can also be solved via its Lagrangian dual, which, after some simplification, reduces to a quadratic program in the dual variables $\alpha_1, \dots, \alpha_n$. The dual is formed via the Karush-Kuhn-Tucker (KKT) conditions [15], which provide a simple means for testing the optimality of a particular solution. In our case, we can use the KKT conditions to express the optimal primal variable $\mathbf{w}$ in terms of the optimal dual variables, according to $\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \Phi(\mathbf{x}_i)$. Note that $\mathbf{w}$ depends only on the $\mathbf{x}_i$ for which $\alpha_i \neq 0$, which are called the *support vectors.* Furthermore, observe that, with this substitution, the quadratic program depends on the training data only through $\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)\rangle_{\mathcal{H}}$ for all possible pairs of training vectors. If we consider a positive semidefinite kernel, i.e., a function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ such that $[k(\mathbf{x}_i, \mathbf{x}_j)]_{ij=1}^n$ is a positive semidefinite matrix for all n and all $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$, then there exists a space $\mathcal{H}$ and a mapping Φ such that $k(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}')\rangle_{\mathcal{H}}$ [14]. By selecting such Φ as the nonlinear feature mapping, we can efficiently compute inner products in $\mathcal{H}$ without explicitly evaluating Φ. In the sequel, we work with positive semidefinite kernels.

To reduce sensitivity to outliers and allow for nonseparable data, it is usually desirable to relax the constraint that each training vector is classified correctly through the introduction of *slack variables*, i.e., we replace the constraints of (P) with $y_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle_{\mathcal{H}} + b) \geq 1 - \xi_i$, where $\xi_i \geq 0$. If $\xi_i > 0$, this means that the corresponding $\mathbf{x}_i$ lies inside the margin and is called a *margin error.* To penalize margin errors while retaining a convex optimization problem, one typically incorporates $\sum_{i=1}^n \xi_i$ into the objective function.

There are two ways to do this, resulting in two SVM formulations. The original SVM adds $C\sum_{i=1}^n \xi_i$ to the objective function, where $C > 0$ is a cost parameter selected by the user; hence, we call this formulation the C-SVM [16]. An alternative (but equivalent) formulation is the ν-SVM [17], which instead adds $\frac{1}{n}\sum_{i=1}^n \xi_i - \nu\rho$ and replaces the constraints with $y_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle_{\mathcal{H}} + b) \geq \rho - \xi_i$, where $\nu \in [0, 1]$ is again a user-supplied parameter and ρ is a variable to be optimized. An advantage of the ν formulation is that ν serves as an upper bound on the fraction of margin errors and a lower bound on the fraction of support vectors [17].

2.2 Cost-Sensitive SVMs

Cost-sensitive extensions of both the C-SVM and the ν-SVM have been proposed—the $2C$-SVM and the 2ν-SVM. We first consider the $2C$-SVM proposed in [7]. Let $I_+ = \{i : y_i = +1\}$ and $I_- = \{i : y_i = -1\}$. The $2C$-SVM quadratic program has primal formulation

$$(P_{2C}) \qquad \min_{\mathbf{w},b,\boldsymbol{\xi}} \ \frac{1}{2}\|\mathbf{w}\|^2 + C\gamma\sum_{i\in I_+}\xi_i + C(1-\gamma)\sum_{i\in I_-}\xi_i$$
$$\text{s.t.} \quad \begin{aligned} y_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle_{\mathcal{H}} + b) \geq 1 - \xi_i, &\quad \text{for } i = 1, \dots, n, \\ \xi_i \geq 0, &\quad \text{for } i = 1, \dots, n, \end{aligned}$$

and simplified Lagrangian dual formulation

$$(D_{2C}) \qquad \min_{\boldsymbol{\alpha}} \ \frac{1}{2}\sum_{i,j=1}^n \alpha_i\alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^n \alpha_i$$
$$\text{s.t.} \quad \begin{aligned} &0 \leq \alpha_i \leq C\gamma, && \text{for } i \in I_+, \\ &0 \leq \alpha_i \leq C(1-\gamma), && \text{for } i \in I_-, \\ &\sum_{i=1}^n \alpha_i y_i = 0, \end{aligned}$$

where $C > 0$ is again a cost parameter set by the user and $\gamma \in [0, 1]$ controls the trade-off between the two types of errors. Note that it is also possible to parameterize the $2C$-SVM through the parameters $C_+ = C\gamma$ and $C_- = C(1-\gamma)$, which is somewhat more common in the literature [7], [8], [9].

As before, one can replace the parameter C with a parameter $\nu \in [0, 1]$ to obtain a cost-sensitive extension of the ν-SVM [6]. The 2ν-SVM has primal

$$
(P_{2\nu}) \qquad \min_{\mathbf{w},b,\boldsymbol{\xi},\rho} \frac{1}{2}\|\mathbf{w}\|^2 - \nu\rho + \frac{\gamma}{n}\sum_{i\in I_+}\xi_i + \frac{1-\gamma}{n}\sum_{i\in I_-}\xi_i
$$

$$
\text{s.t.} \quad \begin{array}{ll} y_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle_{\mathcal{H}} + b) \ge \rho - \xi_i, & \text{for } i = 1, \dots, n, \\ \xi_i \ge 0, & \text{for } i = 1, \dots, n, \\ \rho \ge 0 & \end{array}
$$

and dual

$$
(D_{2\nu}) \qquad \min_{\boldsymbol{\alpha}} \quad \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)
$$

$$
\text{s.t.} \quad \begin{array}{ll} 0 \le \alpha_i \le \dfrac{\gamma}{n}, & \text{for } i \in I_+, \\ 0 \le \alpha_i \le \dfrac{1-\gamma}{n}, & \text{for } i \in I_-, \end{array}
$$

$$
\sum_{i=1}^{n}\alpha_i y_i = 0, \quad \sum_{i=1}^{n}\alpha_i \ge \nu.
$$

As with the $2C$-SVM, the 2ν-SVM has an alternative parameterization. Instead of ν and γ, we can use ν_+ and ν_-. If we let $n_+ = |I_+|$ and $n_- = |I_-|$, then

$$
\nu = \frac{2\nu_+\nu_- n_+ n_-}{(\nu_+ n_+ + \nu_- n_-)n}, \quad \gamma = \frac{\nu_- n_-}{\nu_+ n_+ + \nu_- n_-} = \frac{\nu n}{2\nu_+ n_+},
$$

or equivalently

$$
\nu_+ = \frac{\nu n}{2\gamma n_+}, \quad \nu_- = \frac{\nu n}{2(1-\gamma)n_-}.
$$

This parameterization is more awkward to deal with in establishing the theorems below, but ν_+ and ν_- have a more intuitive meaning than ν and γ, as illustrated below by Proposition 1. Furthermore, Proposition 3 shows that the feasible set of $(D_{2\nu})$ is nonempty if and only if $(\nu_+, \nu_-) \in [0,1]^2$. Thus, this parameterization lends itself naturally toward simple uniform grid searches and a number of additional methods that aid in accurate and efficient parameter selection, as described in Section 3.

2.3 Properties of the 2ν-SVM

Before establishing the relationship between the $2C$-SVM and the 2ν-SVM, we establish some of the basic properties of the 2ν-SVM. We begin by briefly repeating a result of [6] concerning the interpretation of the parameters in the (ν_+, ν_-) formulation.

Proposition 1 [6]. *Suppose that the optimal objective value of $(D_{2\nu})$ is not zero. For the optimal solution of $(D_{2\nu})$, let ME_+ and ME_- denote the fraction of margin errors from classes $+1$ and -1, and let SV_+ and SV_- denote the fraction of support vectors from classes $+1$ and -1. Then,*

$$
\begin{aligned} ME_+ &\le \nu_+ \le SV_+, \\ ME_- &\le \nu_- \le SV_-. \end{aligned}
$$

Returning to the (ν, γ) formulation, we establish the following result concerning the feasibility of $(D_{2\nu})$.

Proposition 2. *Fix $\gamma \in [0,1]$. The feasible set of $(D_{2\nu})$ is nonempty if and only if $\nu \le \nu_{\max} \le 1$, where*

$$
\nu_{\max} = \frac{2\min(\gamma n_+, (1-\gamma)n_-)}{n}.
$$

Proof. First, assume that $\nu \le \nu_{\max}$. Let

$$
\alpha_i = \frac{\nu_{\max}}{2n_+} = \frac{\min(\gamma, (1-\gamma)n_-/n_+)}{n} \le \frac{\gamma}{n}, \; i \in I_+
$$

and

$$
\alpha_i = \frac{\nu_{\max}}{2n_-} = \frac{\min(\gamma n_+/n_-, 1-\gamma)}{n} \le \frac{1-\gamma}{n}, \; i \in I_-.
$$

Then, $\sum_{i\in I_+}\alpha_i + \sum_{i\in I_+}\alpha_i = \nu_{\max} \ge \nu$ and $\sum_{i=1}^{n}\alpha_i y_i = 0$. Thus, $\boldsymbol{\alpha}$ satisfies the constraints of $(D_{2\nu})$ and, hence, $(D_{2\nu})$ is feasible.

Now, assume that $\boldsymbol{\alpha}$ is a feasible point of $(D_{2\nu})$. Then, $\sum_{i=1}^{n}\alpha_i \ge \nu$ and $\sum_{i\in I_+}\alpha_i = \sum_{i\in I_-}\alpha_i$. Combining these, we obtain $\nu \le 2\sum_{i\in I_+}\alpha_i$. Since $0 \le \alpha_i \le \gamma/n$ for $i \in I_+$, we see that $\nu \le 2\sum_{i\in I_+}\alpha_i \le 2\gamma n_+/n$, and therefore, $\nu \le 2\gamma n_+/n$. Similarly, $\nu \le 2(1-\gamma)n_-/n$. Thus, $\nu \le \nu_{\max}$.

Finally, we see that

$$
\nu_{\max} = \frac{2\min(\gamma n_+, (1-\gamma)n_-)}{n} \le \frac{2\min(n_+, n_-)}{n} \le 1,
$$

as desired. □

From Proposition 2, we obtain the following result concerning the (ν_+, ν_-) formulation.

Proposition 3. *The feasible set of $(D_{2\nu})$ is nonempty if and only if $\nu_+ \le 1$ and $\nu_- \le 1$.*

Proof. From Proposition 2, we have that $(D_{2\nu})$ is feasible if and only if

$$
\nu \le \frac{2\min(\gamma n_+, (1-\gamma)n_-)}{n}.
$$

Thus, $(D_{2\nu})$ is feasible if and only if

$$
\frac{2\nu_+\nu_- n_+ n_-}{(\nu_+ n_+ + \nu_- n_-)n} \le \frac{2\min\left(\frac{\nu_- n_+ n_-}{\nu_+ n_+ + \nu_- n_-}, \frac{\nu_+ n_+ n_-}{\nu_+ n_+ + \nu_- n_-}\right)}{n},
$$

and thus, $\nu_+\nu_- \le \min(\nu_-, \nu_+)$ or $\nu_+ \le 1$ and $\nu_- \le 1$. □

2.4 Relationship Between the 2ν-SVM and $2C$-SVM

The following theorems extend the results of [18] and relate (D_{2C}) and $(D_{2\nu})$. The first shows how solutions of (D_{2C}) are related to solutions of $(D_{2\nu})$, and the second shows how solutions of $(D_{2\nu})$ are related to solutions of (D_{2C}). The third theorem, the main result of this section, shows that increasing ν is equivalent to decreasing C. These results collectively establish that (D_{2C}) and $(D_{2\nu})$ are equivalent in that they explore the same set of possible solutions. However, despite their theoretical equivalence, in practice, the 2ν-SVM lends itself toward more effective parameter selection procedures. The theorems and their proofs are inspired by their analogues for (D_C) and (D_ν). However, note that the introduction of the parameter γ somewhat complicates the proofs of these theorems, which are given in the Appendix.

Theorem 1. *Fix $\gamma \in [0,1]$. For any $C > 0$, let $\boldsymbol{\alpha}^C$ be an optimal solution of (D_{2C}) and set $\nu = \sum_{i=1}^{n}\alpha_i^C/(Cn)$. Then, $\boldsymbol{\alpha}$ is an optimal solution of (D_{2C}) if and only if $\boldsymbol{\alpha}/(Cn)$ is an optimal solution of $(D_{2\nu})$.*

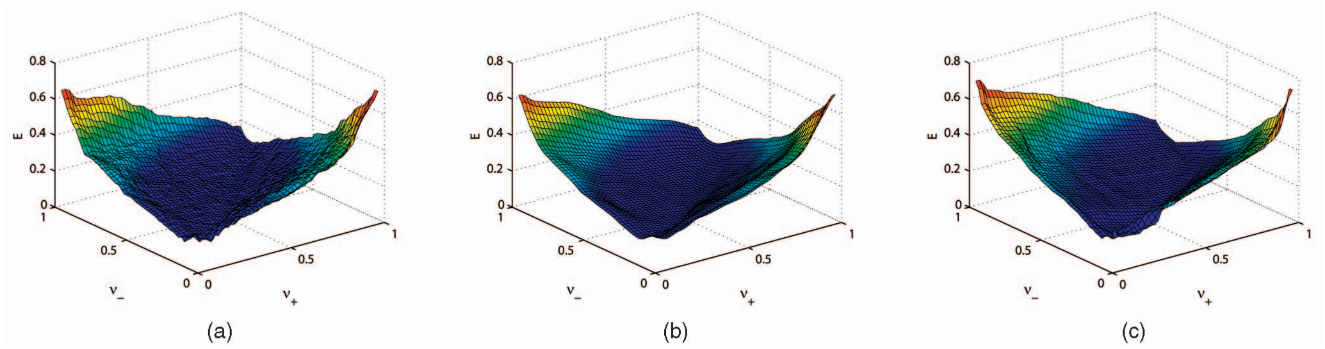


Fig. 1. Effect of 3D smoothing on $\widehat{E}^{CV}_{MM}$ for "banana" data set for $(\nu_+, \nu_-) \in [0,1]^2$. Results are for a representative kernel parameter value. (a) CV estimate: $\widehat{E}^{CV}_{MM}$. (b) Smoothed CV estimate: $\widehat{E}^{SM}_{MM}$. (c) Estimate of E_{MM} based on an independent test set.

Theorem 2. *Fix $\gamma \in [0,1]$. For any $\nu \in (0, \nu_{\max}]$, assume $(D_{2\nu})$ has a nonzero optimal objective value. This implies that the ρ component of an optimal solution of the primal satisfies $\rho > 0$, so we may set $C = 1/(\rho n)$. Then, $\boldsymbol{\alpha}$ is an optimal solution of (D_{2C}) if and only if $\boldsymbol{\alpha}/(Cn)$ is an optimal solution of $(D_{2\nu})$.*

Theorem 3. *Fix $\gamma \in [0,1]$ and let $\boldsymbol{\alpha}^C$ be an optimal solution of (D_{2C}) for all $C > 0$. Define*

$$\nu_* = \lim_{C\to\infty} \frac{\sum_{i=1}^n \alpha_i^C}{Cn}$$

and

$$\nu^* = \lim_{C\to 0} \frac{\sum_{i=1}^n \alpha_i^C}{Cn}.$$

Then, $0 \leq \nu_* \leq \nu^* = \nu_{\max} \leq 1$. For any $\nu > \nu^*$, $(D_{2\nu})$ is infeasible. For any $\nu \in (\nu_*, \nu^*]$ the optimal objective value of $(D_{2\nu})$ is strictly positive, and there exists at least one $C > 0$ such that the following holds: $\boldsymbol{\alpha}$ is an optimal solution of (D_{2C}) if and only if $\boldsymbol{\alpha}/(Cn)$ is an optimal solution of $(D_{2\nu})$). For any $\nu \in [0, \nu_*]$, $(D_{2\nu})$ is feasible with an optimal objective value of zero and a trivial solution.

Remark. Consider the case where the training data can be perfectly separated by a hyperplane in $\mathcal{H}$. In this case, as $C \to \infty$, margin errors are penalized more heavily, and thus for some sufficiently large C, the solution of (D_{2C}) will correspond to a separating hyperplane. Thus, there exists some C^* such that $\boldsymbol{\alpha}^{C^*}$ (corresponding to the separating hyperplane) is an optimal solution of (D_{2C}) for all $C \geq C^*$. In this case, as $C \to \infty$, $\sum_{i=1}^n \alpha_i^C / Cn \to 0$, and thus, $\nu_* = 0$. Note also that we can easily restate Theorem 3 for the alternative (C_+, C_-) and (ν_+, ν_-) parameterizations if desired.

3 Support Vector Algorithms for Minimax and NP Classification

In order to apply either the $2C$-SVM or the 2ν-SVM to the problems of minimax or NP classification, we must set the free parameters appropriately. In light of Theorem 3, it might appear that it makes no difference which formulation we use, but given the critical importance of parameter selection to both of these problems, any practical advantage that one parametrization offers over the other is extremely important. In our case, we are motivated to employ the 2ν-SVM for two reasons. First, the $2C$-SVM has an unbounded parameter space. In our experience, this leads to numerical issues for very large or small parameter values, and it also entails a certain degree of arbitrariness in selecting the starting and ending search grid points. Since the parameter space of the 2ν-SVM is bounded, we can conduct a simple uniform grid search over $[0,1]^2$ to select (ν_+, ν_-). The second reason is that we have found a method, described below, that capitalizes on this uniform grid to significantly enhance the accuracy of error estimates for the 2ν-SVM.

To select the appropriate (ν_+, ν_-), we obtain estimates of the error rates over a grid of possible parameter values and select the best parameter combination based on these estimates. The central focus of our study (which will be based on simulations across a wide range of data sets) is concerned with how to most accurately and efficiently perform this error estimation and parameter selection process.

To be concrete, we will describe the algorithm for the radial basis function (Gaussian) kernel, although the method could easily be adapted to other kernels. We consider a 3D grid of possible values for ν_+, ν_-, and the kernel bandwidth parameter σ. For each possible combination of parameters, we begin by obtaining CV estimates of the false alarm and miss rates, which we denote $\widehat{P}^{CV}_F$ and $\widehat{P}^{CV}_M$. Note that we slightly abuse notation and that $\widehat{P}_F$ and $\widehat{P}_M$ should be thought of as arrays indexed by ν_+, ν_-, and σ. (This is distinct from the notation established earlier where P_F and P_M are functionals that map classifiers to error rates.) We next select the parameter combination that minimizes $\widehat{E}^{CV}$, where for minimax classification, we set $\widehat{E}^{CV} = \widehat{E}^{CV}_{MM} = \max\{\widehat{P}^{CV}_F, \widehat{P}^{CV}_M\}$ and for NP classification, we set $\widehat{E}^{CV} = \widehat{E}^{CV}_{NP(\alpha)}$, where $\widehat{E}^{CV}_{NP(\alpha)} = \widehat{P}^{CV}_M$ when $\widehat{P}^{CV}_F \leq \alpha$ and $\widehat{E}^{CV}_{NP(\alpha)} = \infty$ otherwise.

3.1 Accurate Error Estimation: Smoothed Cross-Validation

While CV estimates are relatively easy to calculate, they tend to have a high variance, and hence, some parameter combinations will look much better than they actually are due to chance variation. However, we have observed across a wide range of data sets for the 2ν-SVM that $\widehat{P}^{CV}_F$ and $\widehat{P}^{CV}_M$ appear to somewhat "noisy" versions of smoothly varying functions of (ν_+, ν_-, σ), as illustrated in Fig. 1a. This motivates a simple heuristic to improve upon CV: Smooth $\widehat{P}^{CV}_F$ and $\widehat{P}^{CV}_M$ through convolution with a low-pass filter W and then calculate $\widehat{E}^{SM}$ using the smoothed CV estimates. Ignoring the kernel parameter, we describe the approach in Algorithm 1. We also consider two approaches to selecting the kernel parameter. We can apply a two-dimensional (2D) filter to the error estimates for $(\nu_+, \nu_-) \in [0,1]^2$ as in Algorithm 1, separately for each value of σ, or alternatively, a three-dimensional (3D) filter to the error estimates, smoothing across different kernel parameter values. Fig. 1 illustrates the effect of 3D smoothing on an example data set, demonstrating that $\widehat{E}^{SM}$ more closely resembles the estimate of E obtained from an independent test set. In our experiments, the filter is chosen to be a simple Gaussian window low-pass filter. Several possible filters can be used (for example, Gaussian filters of varying widths, median filters, etc.), and all result in similar performance gains. The key to all of these smoothing approaches is that they perform some kind of local averaging to reduce outlying estimates. We will see that both 2D and 3D methods are extremely effective in a quantitative sense in Section 5.

```
Algorithm 1. Smoothed Grid Search
  for a vector of values of $\nu_+$ do
      for a vector of values of $\nu_-$ do
          $\widehat{E}^{CV} \leftarrow$ CV estimate of $E$
      end for
  end for
  $\widehat{E}^{SM} \leftarrow W(\widehat{E}^{CV})$
  select $\nu_+$, $\nu_-$ minimizing $\widehat{E}^{SM}$
  train SVM using $\nu_+$, $\nu_-$
```

3.2 Efficient and Accurate Error Estimation: Coordinate Descent

The additional parameter in the 2ν-SVM can render a full grid search, somewhat computationally expensive, especially for large data sets. Fortunately, a simple speedup heuristic exists. Again inspired by the smoothness of $\widehat{P}_F^{CV}$ and $\widehat{P}_M^{CV}$, we propose a coordinate descent search. Several variants are possible, but the simplest one we employ, denoted as 2D coordinate descent, is described in Algorithm 2. It essentially consists of a sequence of orthogonal line searches that continues until it converges to a fixed point. To incorporate a kernel parameter, we can either repeat this approach for each value of the kernel parameter, or consider the natural 3D extension of this algorithm. Smoothing can also be easily incorporated into this framework by conducting "tube searches": adding additional adjacent line searches adjacent to the line searches in Algorithm 2 that are then filtered to yield smoothed estimates along the original line searches.

```
Algorithm 2. Coordinate Descent
  $(\nu_+^0, \nu_-^0) \leftarrow (0.5, 0.5)$
  $i \leftarrow 0$
  repeat
      estimate $E$ for $\nu_+ = \nu_+^i$ and a vector of values of $\nu_-$
      estimate $E$ for $\nu_- = \nu_-^i$ and a vector of values of $\nu_+$
      set $\nu_+^{i+1}$, $\nu_-^{i+1}$ to minimize $\widehat{E}^{CV}$
      increment $i$
  until $\nu_+^i = \nu_+^{i-1}$ and $\nu_-^i = \nu_-^{i-1}$
  train SVM using $\nu_+^i$, $\nu_-^i$
```

4 Experimental Setup

4.1 Performance Evaluation

In order to evaluate the methods described above and to compare the 2ν-SVM to methods more commonly used in practice, we conduct a detailed experimental study. We compare the algorithms on a collection of 11 benchmark data sets representing a variety of dimensions and sample sizes.[1] The data sets comprise a mixture of synthetic and real data. For each of the first nine data sets, we have 100 permutations of the data into training and test sets, and for the last two, we have 20 permutations. We use the different permutations to generate a more reliable performance estimate for each algorithm. For a given algorithm, we train a classifier for each permutation of training data and then evaluate our performance metric using the corresponding permutation of the test data. We then average the scores over all permutations. Specifically, for each approach, we estimate $\widehat{P}_F^{CV}$ and $\widehat{P}_M^{CV}$ for various parameter combinations using five-fold CV. We then select the appropriate parameters, retrain our classifiers on the full set of training data, and then estimate $P_F(f)$ and $P_M(f)$ using the independent test data.

Our performance metric is $\max\{P_F(f), P_M(f)\}$ for minimax classification. For NP classification, we use the *Neyman-Pearson score*,

$$\frac{1}{\alpha}\max\{P_F(f) - \alpha, 0\} + P_M(f), \tag{6}$$

proposed in [19]. It can be shown that the global minimizer of (6) is the optimal NP classifier under general conditions on the underlying distribution. Furthermore, the NP score has additional properties, desirable from a statistical point of view: It can be reliably estimated from data, it tolerates small violations of the false alarm constraint, and as α draws closer to zero, a stiffer penalty is exacted on classifiers that violate the constraint [19]. To evaluate performance on unbalanced data sets, we repeated these experiments, retaining only 10 percent of the negatively labeled training data.

In order to compare multiple algorithms on multiple data sets, we use the two-step procedure advocated in [20]. First, we use the Friedman test, a statistical test for determining whether the observed differences between the algorithms are statistically significant. When reporting results from the Friedman test, we give the p-value. Next, once we have rejected the null hypothesis (that the differences have occurred by chance), we apply the Nemenyi test, which involves computing a ranking of the algorithms for each data set, and then an average ranking for each algorithm. Along with these rankings, we provide the so-called critical difference for a significance level of 0.05. (If the average ranking of two algorithms differs by more than this value, which depends on the desired p-value and the number of algorithms being compared against each other, then the performance of the two algorithms is significantly different, with a p-value of at most 0.05.) See [20] for a more thorough discussion of and motivation for these techniques.

[1] We use the following data sets, which can be obtained with documentation from http://ida.first.fhg.de/projects/bench: banana, breast-cancer, diabetes, flare-solar, heart, ringnorm, thyroid, twonorm, waveform, image, splice.

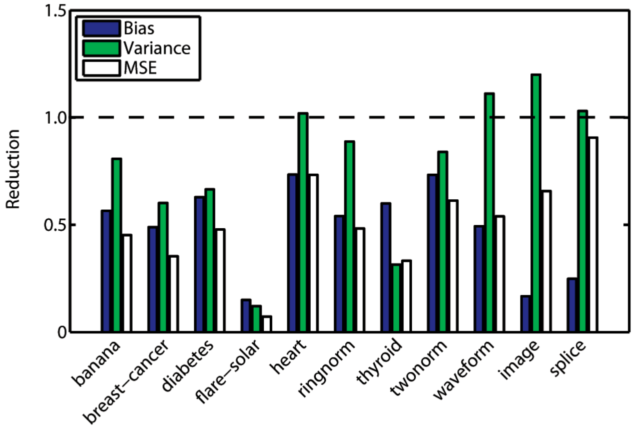


Fig. 2. Effect of smoothing on $\widehat{E}_{MM}^{CV}$. The results shown are the ratio of the bias, variance, and mean squared error (MSE) of $\widehat{E}_{MM}^{SM}$ to that of $\widehat{E}_{MM}^{CV}$ for each data set. A value of less than 1 indicates improvement.

4.2 Implementation

In all experiments we use a radial basis function (Gaussian) kernel, a logarithmically spaced grid of 50 points of $\sigma \in [10^{-4}, 10^4]$, and a 50×50 regular grid of $(\nu_+, \nu_-) \in [0,1]^2$. For the 2D smoothing approach, we apply a 3×3 Gaussian window to the error estimates for $(\nu_+, \nu_-) \in [0,1]^2$ separately for each value of σ. For the 3D smoothing approach, we apply a $3 \times 3 \times 3$ Gaussian window to the error estimates, smoothing across different kernel parameter values. The standard deviation of the Gaussian window is set to the length of one grid interval. (There does not seem to be much change in performance for different window sizes and widths.)

Our implementation of the 2ν-SVM uses the sequential minimization optimization (SMO) approach. The idea of the SMO algorithm is to break up the optimization problem by iteratively selecting pairs (α_i, α_j) and then optimizing the objective function with respect to (α_i, α_j) while holding the remaining α_k constant. This subproblem has an analytic solution, and hence, can be solved extremely efficiently. The algorithm then proceeds by iteratively selecting pairs of variables to optimize (usually according to a criterion based on the violation of the KKT constraints). For a detailed discussion of the SMO algorithm as applied to the ν-SVM, see [21]. A key point noted in [21] is that optimizing over a particular pair (α_i, α_j) will only reduce the objective function if $y_i = y_j$. This means that an SMO-type implementation of the 2ν-SVM will only differ from that of the ν-SVM in that we must replace the optimization constraint that $\alpha_i, \alpha_j \in [0, 1/n]$ with $\alpha_i, \alpha_j \in [0, \gamma/n]$ for $i \in I_+$ and $\alpha_i, \alpha_j \in [0, (1-\gamma)/n]$ for $i \in I_+$. The remaining steps of the algorithm, including the subset selection methods, are identical to those of the ν-SVM. Our implementation is based on the popular LIBSVM package [13]. Our code, as well as a more detailed discussion of the changes made, are available online at www.dsp.rice.edu/software.

4.3 Alternative Approaches to Controlling Errors

In order to provide a reference for comparison, we also consider two alternative SVM-based approaches to controlling P_F and P_M, *bias-shifting* and the *balanced ν-SVM*. In bias-shifting, which is the most common approach taken in the literature, we train a standard (cost-insensitive) SVM and then adjust the bias of the resulting classifier to achieve the desired error rates [22]. Note that we do not expect that bias-shifting will perform as well as the 2ν-SVM since it has been shown that the cost-sensitive SVM is superior to bias-shifting in the sense that it will generate an ROC with a larger area under its curve [22]. In our experiments, we search over a uniform grid of 50 points of the parameter ν and also apply a 3×3 Gaussian smoothing filter to smooth the error estimates across different values of ν and σ.

TABLE 1
Average Ranking of Each Smoothing Approach for the 2ν-SVM

	Smoothing	Balanced	Unbalanced
Minimax	None	2.91	2.91
	2-D	1.73	1.64
	3-D	**1.36**	**1.45**
NP	Smoothing	Balanced	Unbalanced
	None	2.73	2.64
	2-D	2.09	2.00
	3-D	**1.18**	**1.36**

Friedman p-values are <0.01 for all cases; Nemenyi critical difference at 0.05 is 1.10.

A common motivation for minimax classification is that some data sets are unbalanced in the sense that they have many more samples from one class than from the other. In light of Proposition 1, another possible algorithm is to use a 2ν-SVM with $\nu_+ = \nu_-$. We refer to this method as the *balanced ν-SVM*. Since ν_+ and ν_- are upper bounds on the fractions of margin errors from their respective classes, we might expect that this method will be superior to the traditional ν-SVM for minimax classification. Note that this method has the same computational complexity as the traditional ν-SVM. For the balanced ν-SVM, we search over a uniform grid of 50 points of the parameter $\nu_+ = \nu_-$ and again apply a 3×3 Gaussian smoothing filter to smooth the error estimates across different σ.

5 Results and Discussion

5.1 Effects of Smoothing

In Fig. 2, we examine how smoothing impacts the accuracy of the error estimates for each of our data sets. We compare the CV error estimates and the test error estimates for the parameter combination selected using the CV estimates. We then repeat this for smoothed error estimates. We compute the bias, variance, and mean squared error (MSE) of the two estimation approaches by averaging over different permutations. From Fig. 2, we see that smoothing leads to significant reductions in the bias and MSE across all data sets. On most of the data sets, we also observe a reduction in the variance. Furthermore, while we do not display the actual values, we also note that the bias of the CV estimator is always negative and ranges from -0.01 to as large as -0.17. This validates our intuition that the "noise" in the CV estimates can lead to selecting parameter combinations that look better than they really are. The bias, variance, and MSE reductions translate into a drastic improvement on the resulting classifiers. The results of smoothing on our benchmark data sets are shown in Table 1, and they clearly

TABLE 2
Average Ranking of Each Coordinate Descent Approach for the 2ν-SVM

	Smoothing	CD	Balanced	Unbalanced
Minimax	None	2-D	4.18	4.18
	None	3-D	3.91	4.00
	2-D	2-D	2.73	2.82
	3-D	2-D	**2.00**	**2.00**
	3-D	3-D	2.18	**2.00**
	Smoothing	CD	Balanced	Unbalanced
NP	None	2-D	3.82	4.36
	None	3-D	3.55	3.64
	2-D	2-D	2.64	3.36
	3-D	2-D	**1.91**	1.91
	3-D	3-D	3.09	**1.73**

Friedman p-values are <0.05 for all cases; Nemenyi critical difference at 0.05 is 1.92.

TABLE 3
Average Ranking of the 2ν-SVM Methods, the Balanced ν-SVM, and the ν-SVM with Bias-Shifting

	Method	Balanced	Unbalanced
Minimax	3D-SGS	2.73	**2.00**
	2D-CD	**2.64**	2.64
	3D-CD	2.73	**2.00**
	ν-SVM	3.64	4.09
	Bal ν-SVM	3.27	4.27
	Method	Balanced	Unbalanced
NP	3D-SGS	2.36	3.18
	2D-CD	**2.18**	2.09
	3D-CD	2.73	**1.64**
	ν-SVM	4.91	4.18
	Bal ν-SVM	2.82	3.91

Friedman p-values are <0.001 for all cases except unbalanced minimax classification, for which the p-value is 0.502; Nemenyi critical difference at 0.05 is 1.92.

indicate that both 2D and 3D smoothing offer a statistically significant gain in performance, with 3D smoothing offering a slight edge.

5.2 Coordinate Descent

Table 2 shows that 3D smoothing combined with either 2D or 3D coordinate descent offers gains in performance as well, which is particularly helpful since these methods speedup the parameter selection process considerably. Note that smoothing again makes a tremendous impact on the resulting performance, even in the absence of a complete grid search. Perhaps somewhat surprisingly, we observe that 2D and 3D coordinate descent behave similarly, despite 3D coordinate descent being considerably more greedy.

5.3 Comparison with Other Methods

We now compare the 2ν-SVM strategies to the balanced ν-SVM and traditional ν-SVM with bias-shifting. Table 3 provides the results of the Nemenyi test for the 3D smoothed grid-search approach (labeled 3D-SGS), the 2D and 3D coordinate descent methods (labeled 2D-CD and 3D-CD—both use 3D smoothing), the balanced ν-SVM without bias-shifting (labeled Bal ν-SVM), and the traditional ν-SVM with bias-shifting (labeled ν-SVM). In Table 4, we compare the training times for these methods. Since there is a large variation in training time across the different data sets, we normalize the training time by the training time of the 3D smoothed grid search. The values listed are the average improvement (across the different permutations) over the 3D smoothed grid search achieved by the different approaches. We report the results for minimax classification; the results for NP classification across the different values of α are very similar.

For the case of minimax classification on balanced data sets, the 2ν-SVM methods appear to exhibit stronger performance, but this is not statistically significant. However, for the *un*balanced case, there is a clear and significant difference, with the 2ν-SVM methods being clearly superior. The 3D-SGS method appears to be the best performing overall, but the coordinate descent methods exhibit very similar performance. For the case of NP classification, the 2ν-SVM methods clearly outperform the traditional ν-SVM methods and also outperform the balanced ν-SVM.

As expected, the 3D-SGS tends to take on the order of 50 times longer to train compared to the ν-SVM and Bal

TABLE 4
Speedup in Training Time for the 2ν-SVM Methods for Minimax Classification

Method	banana	breast-cancer	diabetes	flare-solar	heart	ringnorm	thyroid	twonorm	waveform	image	splice
3D-SGS	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2D-CD	3.33	4.61	5.17	3.78	5.47	5.46	4.69	6.48	5.86	3.40	5.41
3D-CD	**53.60**	34.78	48.80	48.32	42.78	44.63	60.10	**49.33**	32.05	34.74	32.32
ν-SVM	45.05	**74.06**	**58.96**	**52.12**	**50.51**	**48.99**	**73.84**	48.81	**74.77**	**56.40**	**51.87**
Bal ν-SVM	43.49	68.49	56.31	50.62	48.49	47.54	69.61	47.10	71.31	54.17	49.27

The reported values are the average improvement (across the different permutations) over the 3D-SGS approach for each method and each data set. (A value of 50 indicates that a method was 50 times faster than the 3D-SGS approach on that data set.)

ν-SVM (as a result of having to collect CV estimates over a 50×50 grid of values for (ν_+, ν_-) instead of a length of 50 grid of values for ν). However, the coordinate descent methods offer a large improvement over the 3D-SGS approach in terms of training time, with little loss in performance. In particular, the 2D-CD approach results in training times that are roughly five times faster than the 3D-SGS approach (although still 10 times slower than the ν-SVM and Bal ν-SVM), while the 3D-CD approach requires a training time on the same order as the ν-SVM and Bal ν-SVM. On occasion, the 3D-CD approach is even faster than the ν-SVM and Bal ν-SVM. Thus, we would recommend the 3D-CD approach as a suitable balance between accuracy and computational efficiency.

Perhaps the most surprising result is that the 3D coordinate descent method is not only competitive with the full grid search but even performs better than the grid search on the unbalanced data sets. This may be a consequence of the fact that, by ignoring many parameter combinations, coordinate descent is less sensitive to noisy error estimates. In essence, coordinate descent can act as a simple form of complexity regularization, thus preventing overfitting.

6 Conclusion

We have demonstrated that, when learning with respect to the minimax or NP criteria, the 2ν-SVM, in conjunction with smoothed cross-validation error estimates, clearly outperforms methods based on raw (unsmoothed) error estimates, as well as the bias-shifting strategies commonly used in practice. Our approach exploits certain properties of the 2ν-SVM and its parameter space, which we analyzed and related to the $2C$-SVM. Our experimental results imply that accurate error estimation is crucial to our algorithm's performance. Simple smoothing techniques lead to significantly improved error estimates, which translate into better parameter selection and a dramatic improvement in performance. We have also illustrated a computationally efficient variant of our approach based on coordinate descent.

The primary intuition explaining the gains achieved by our approach lie in minimizing the impact of outlying error estimates. When estimating errors for a large grid of parameter values, a poor estimator is likely to be overly optimistic at a few parameter settings simply by chance. Our smoothing approach performs a weighted local averaging to reduce outlying estimates. This may also explain the surprising performance of our greedy coordinate descent speedup: By ignoring many parameter combinations, the algorithm reduces its exposure to such outliers.

Appendix

In [18], Chang and Lin illustrate the relationship between (D_ν) and (D_C)—which denote the dual formulations of the ν-SVM and C-SVM, respectively. We follow a similar course. First, we rescale (D_{2C}) by Cn in order to compare it with $(D_{2\nu})$. This yields:

$$(D'_{2C}) \qquad \min_{\boldsymbol{\alpha}} \quad \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) - \frac{1}{Cn}\sum_{i=1}^{n} \alpha_i$$

$$\text{s.t.} \quad \begin{aligned} & 0 \le \alpha_i \le \frac{\gamma}{n}, \qquad i \in I_+, \\ & 0 \le \alpha_i \le \frac{1-\gamma}{n}, \quad i \in I_-, \\ & \sum_{i=1}^{n} \alpha_i y_i = 0. \end{aligned}$$

In order to prove the theorems in Section 2.4, we take advantage of the equivalence of (D_{2C}) and (D'_{2C}). We will establish the relationship between $(D_{2\nu})$ and (D'_{2C}), which by rescaling establishes the theorems in Section 2.4 relating $(D_{2\nu})$ and (D_{2C}). We begin with the following lemmata:

Lemma 1. *Fix $\gamma \in [0,1]$ and $\nu \in [0, \nu_{\max}]$. There is at least one optimal solution of $(D_{2\nu})$ that satisfies $\sum_{i=1}^n \alpha_i = \nu$. In addition, if the optimal objective value of $(D_{2\nu})$ is not zero, then all optimal solutions of $(D_{2\nu})$ satisfy $\sum_{i=1}^n \alpha_i = \nu$.*

Proof. The first statement follows from Proposition 3. The second statement was proven in Theorem 1 of [18] for the ν-SVM. The proof relies only upon the form of the objective function of the dual formulation of the ν-SVM, which is identical to that of $(D_{2\nu})$, and the fact that any feasible point can be rescaled so that $\sum_{i=1}^n \alpha_i = \nu$. Thus, we omit it for the sake of brevity and refer the reader to [18]. □

Lemma 2. *Fix $\gamma \in [0,1]$, $C > 0$, and $\nu \in [0,1]$. Assume (D'_{2C}) and $(D_{2\nu})$ share one optimal solution $\boldsymbol{\alpha}^C$ with $\sum_{i=1}^n \alpha_i^C = \nu$. Then, $\boldsymbol{\alpha}$ is an optimal solution of (D'_{2C}) if and only if it is an optimal solution of $(D_{2\nu})$.*

Proof. The analogue of this lemma for (D'_C) and (D_ν) is proven in Lemma 2 of [18]. The proof relies only upon the form of the objective functions, which are identical to those of (D'_{2C}) and $(D_{2\nu})$, on the fact that the feasible sets are convex, and on the analogue of Lemma 1. Thus, we again refer the reader to [18]. □

For the proofs of Theorems 1 and 2, we will employ the Karush-Kuhn-Tucker (KKT) conditions [15]. As noted above, these conditions typically depend on both the primal and dual variables, but in our case, we can eliminate $\mathbf{w}$ to form a simplified set of conditions. Specifically, $\boldsymbol{\alpha}$ is an optimal solution of (D'_{2C}) if and only if there exist $b \in \mathbb{R}$ and $\boldsymbol{\lambda}, \boldsymbol{\xi} \in \mathbb{R}^n$ satisfying the conditions:

$$\sum_{j=1}^{n} \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) - \frac{1}{Cn} + b y_i = \lambda_i - \xi_i \quad \forall\, i, \tag{7}$$

$$\lambda_i \alpha_i = 0, \quad \lambda_i \ge 0, \quad \xi_i \ge 0 \quad \forall\, i, \tag{8}$$

$$\xi_i\left(\frac{\gamma}{n} - \alpha_i\right) = 0, \quad 0 \le \alpha_i \le \frac{\gamma}{n} \qquad i \in I_+, \tag{9}$$

$$\xi_i\left(\frac{1-\gamma}{n} - \alpha_i\right) = 0, \quad 0 \le \alpha_i \le \frac{1-\gamma}{n} \quad i \in I_-, \tag{10}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0. \tag{11}$$

Similarly, $\boldsymbol{\alpha}$ is an optimal solution of $(D_{2\nu})$ if and only if there exist $b, \rho \in \mathbb{R}$ and $\boldsymbol{\lambda}, \boldsymbol{\xi} \in \mathbb{R}^n$ satisfying:

$$\sum_{j=1}^{n} \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \; - \; \rho + b y_i = \lambda_i - \xi_i \quad \forall\, i, \tag{12}$$

$$\lambda_i \alpha_i = 0, \quad \lambda_i \geq 0, \quad \xi_i \geq 0 \quad \forall\, i, \tag{13}$$

$$\xi_i \left(\frac{\gamma}{n} - \alpha_i \right) = 0, \quad 0 \leq \alpha_i \leq \frac{\gamma}{n} \quad i \in I_+, \tag{14}$$

$$\xi_i \left(\frac{1-\gamma}{n} - \alpha_i \right) = 0, \quad 0 \leq \alpha_i \leq \frac{1-\gamma}{n} \quad i \in I_-, \tag{15}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0, \quad \sum_{i=1}^{n} \alpha_i \geq \nu, \quad \rho \left(\sum_{i=1}^{n} \alpha_i - \nu \right) = 0. \tag{16}$$

Note that the two sets of conditions are mostly identical, except for the first and last two of the conditions for $(D_{2\nu})$. Using this observation, we can prove Theorems 1 and 2.

Proof of Theorem 1. If $\boldsymbol{\alpha}^C$ is an optimal solution of (D'_{2C}), then it is a KKT point of (D'_{2C}). By setting $\nu = \sum_{i=1}^{n} \alpha_i^C$ and $\rho = 1/(Cn)$, we see that $\boldsymbol{\alpha}^C$ also satisfies the KKT conditions for $(D_{2\nu})$ and thus is an optimal solution of $(D_{2\nu})$. From Lemma 2, we therefore have that $\boldsymbol{\alpha}$ is an optimal solution of (D'_{2C}) if and only if it is an optimal solution of $(D_{2\nu})$. Thus, $\boldsymbol{\alpha}$ is an optimal solution of (D_{2C}) if and only if $\boldsymbol{\alpha}/(Cn)$ is an optimal solution of $(D_{2\nu})$. □

Proof of Theorem 2. If $\boldsymbol{\alpha}^\nu$ is an optimal solution of $(D_{2\nu})$, then it is a KKT point of $(D_{2\nu})$. From (12), we have

$$\sum_{i=1}^{n} \left(\sum_{j=1}^{n} \alpha_j^\nu y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) - \rho + b y_i \right) \alpha_i^\nu = \sum_{i=1}^{n} (\lambda_i - \xi_i) \alpha_i^\nu,$$

which, by applying (13) and (14), reduces to

$$\sum_{i,j=1}^{n} \alpha_i^\nu \alpha_j^\nu y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) - \rho \sum_{i=1}^{n} \alpha_i^\nu = -\frac{\gamma}{n} \sum_{i=1}^{n} \xi_i.$$

By assumption, $(D_{2\nu})$ has a nonzero optimal objective value. Thus, from Lemma 1, $\sum_{i=1}^{n} \alpha_i^\nu = \nu$ and

$$\rho = \frac{1}{\nu} \left(\sum_{i,j=1}^{n} \alpha_i^\nu \alpha_j^\nu y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) + \frac{\gamma}{n} \sum_{i=1}^{n} \xi_i \right) > 0.$$

Thus, we can choose $C = 1/(\rho n) > 0$ so that $\boldsymbol{\alpha}^\nu$ is a KKT point of (D'_{2C}). From Lemma 2, we have that $\boldsymbol{\alpha}$ is an optimal solution of (D'_{2C}) if and only if it is an optimal solution of $(D_{2\nu})$. Hence, $\boldsymbol{\alpha}$ is an optimal solution of (D_{2C}) if and only if $\boldsymbol{\alpha}/(Cn)$ is an optimal solution of $(D_{2\nu})$. □

We will need the following lemmata to prove Theorem 3.

Lemma 3. *Fix $\gamma \in [0, 1]$ and $\nu \in [0, 1]$. If the optimal objective value of $(D_{2\nu})$ is zero and there is a $C > 0$ such that the optimal solution of (D'_{2C}), $\boldsymbol{\alpha}^C$, satisfies $\sum_{i=1}^{n} \alpha_i^C = \nu$, then $\nu = \nu_{\max}$ and any $\boldsymbol{\alpha}$ is an optimal solution of $(D_{2\nu})$ if and only if it is an optimal solution of (D'_{2C}) for all $C > 0$.*

Proof. Setting $\rho = 1/Cn$, $\boldsymbol{\alpha}^C$ is a KKT point of $(D_{2\nu})$. Hence, if the optimal objective value of $(D_{2\nu})$ is zero, then $\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i^C \alpha_j^C y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) = 0$. The kernel k is (by definition) positive definite, so we have $\sum_{j=1}^{n} \alpha_j^C y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) = 0$. Thus, (7) and (12) become

$$-\frac{1}{Cn} + b y_i = \lambda_i - \xi_i, \quad \text{for} \quad i = 1, \ldots, n. \tag{17}$$

From this, we observe that if $b \geq 0$, then $\lambda_i - \xi_i < 0$ for all $i \in I_-$, and that if $b \leq 0$, then $\lambda_i - \xi_i < 0$ for all $i \in I_+$.

Without loss of generality, we assume that $b \geq 0$ since the situation when $b \leq 0$ can be treated similarly by exchanging I_- and I_+. Since $b > 0$ we have that $\lambda_i - \xi_i < 0$ for all $i \in I_-$, and since the λ_i are nonnegative, this implies that $\xi_i > 0$ for all $i \in I_-$. Therefore, in order for the first conditions of (10) and (15) to hold, we need $\alpha_i^C = (1-\gamma)/n$ for all $i \in I_-$. From the first conditions of (11) and (16), we have that $\sum_{i \in I_+} \alpha_i^C = \sum_{i \in I_-} \alpha_i^C$, and thus $\sum_{i \in I_+} \alpha_i^C = (1-\gamma) n_-/n \leq \gamma n_+/n$.

Thus, for the case where $b \geq 0$, we have established that $\alpha_i^C = (1-\gamma)/n$ for all $i \in I_-$ and that $(1-\gamma)n_- \leq \gamma n_+$. We now consider $i \in I_+$. There are three possibilities, which follow from (17) and depend on b:

1. If $b \in [0, \frac{1}{Cn})$, then $\lambda_i - \xi_i < 0$ for all $i \in I_+$.
2. If $b > \frac{1}{Cn}$, then $\lambda_i - \xi_i > 0$ for all $i \in I_+$.
3. If $b = \frac{1}{Cn}$, then $\lambda_i - \xi_i = 0$ for all $i \in I_+$.

In Case 1, we must have $\xi_i > 0$ for all $i \in I_+$. For the first conditions of (9) and (14) to hold, we need $\alpha_i^C = \gamma/n$ for all $i \in I_+$. The requirement that $\sum_{i \in I_+} \alpha_i^C = \sum_{i \in I_-} \alpha_i^C$ (from the first conditions of (11) and (16)) and the fact that $\alpha_i^C = (1-\gamma)/n$ for all $i \in I_-$ imply that

$$\sum_{i=1}^{n} \alpha_i^C = 2n_+\gamma/n = 2n_-(1-\gamma)/n = \nu_{\max}.$$

Furthermore, since the optimal objective value of $(D_{2\nu})$ is zero, the objective function for (D'_{2C}) in this case becomes

$$\min_{\boldsymbol{\alpha}} \quad -\frac{1}{Cn} \sum_{i=1}^{n} \alpha_i.$$

This is minimized by $\boldsymbol{\alpha}^C$ (since $\sum_{i=1}^{n} \alpha_i^C = \nu_{\max}$), hence, $\boldsymbol{\alpha}^C$ is an optimal solution of (D'_{2C}) for all $C > 0$.

In Case 2, $\lambda_i > 0$ for all $i \in I_-$. For the first conditions of (8) and (13), $\lambda_i \alpha_i^C = 0$, to hold, we need $\alpha_i^C = 0$ for all $i \in I_+$. However, the requirement that $\sum_{i \in I_+} \alpha_i^C = \sum_{i \in I_-} \alpha_i^C$ and the fact that $\alpha_i^C = (1-\gamma)/n$ for all $i \in I_-$ lead to a contradiction if I_- is nonempty. Hence, all of the training vectors are in the same class, and $\alpha_i^C = 0$ for all i. Thus,

$$\sum_{i=1}^{n} \alpha_i^C = 0 = \nu_{\max}.$$

Furthermore, if all the data are from the same class, then $\boldsymbol{\alpha}^C = \mathbf{0}$ is an optimal solution of (D'_{2C}) for all $C > 0$.

In Case 3, where $\lambda_i - \xi_i = 0$, either $\lambda_i = \xi_i \neq 0$ or $\lambda_i = \xi_i = 0$ for each $i \in I_+$. However, $\lambda_i = \xi_i \neq 0$ leads to a contradiction because (8) and (13), together with (9) and (14), require both $\alpha_i^C = 0$ and $\alpha_i^C = \gamma/n$. Thus, $\lambda_i = \xi_i = 0$ and the KKT conditions involving λ_i and ξ_i impose no conditions on α_i^C for $i \in I_+$. Since $\alpha_i^C = (1-\gamma)/n$ for all $i \in I_-$, and $(1-\gamma)n_- \le \gamma n_+$, we can satisfy

$$\sum_{i \in I_+} \alpha_i^C = \sum_{i \in I_-} \alpha_i^C = (1-\gamma)n_+/n.$$

Thus, $\sum_{i=1}^n \alpha_i^C = \nu_{\max}$. Hence, by setting $b = 1/(Cn)$, $\boldsymbol{\alpha}^C$ is an optimal solution of (D'_{2C}) for all $C > 0$.

Therefore, in all three cases, we have that $\nu = \nu_{\max}$ and that $\boldsymbol{\alpha}^C$ is an optimal solution of (D'_{2C}) for all $C > 0$. Hence, if $\boldsymbol{\alpha}^C$ is an optimal solution of (D'_{2C}) and for $\nu = \sum_{i=1}^n \alpha_i^C$ the optimal objective value of $(D_{2\nu})$ is zero, then $\nu = \nu_{\max}$ and $\boldsymbol{\alpha}^C$ is an optimal solution of (D'_{2C}), for all $C > 0$. The lemma follows by combining this with Lemma 2. □

Lemma 4. *If $\boldsymbol{\alpha}^C$ is an optimal solution of (D'_{2C}), then $\sum_{i=1}^n \alpha_i^C$ is a continuous decreasing function of C on $(0, \infty)$.*

Proof. The analogue of this lemma for (D'_C) is proven in [18]. Since the proof depends only on the form of the objective function and the analogues of Theorems 1 and 2 and Lemma 3, we omit the proof and refer the reader to [18]. □

We are now ready to prove the main theorem.

Proof of Theorem 3. From Lemma 4 and the fact that, for all C, $0 \le \sum_{i=1}^n \alpha_i^C \le \nu_{\max}$, we know that the above limits are well defined and exist.

For any optimal solution of (D'_{2C}), (7) holds:

$$\sum_{j=1}^n \alpha_j^C y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) - \frac{1}{Cn} + b = \lambda_i - \xi_i, \quad \text{for} \quad i \in I_+,$$
$$\sum_{j=1}^n \alpha_j^C y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) - \frac{1}{Cn} - b = \lambda_i - \xi_i, \quad \text{for} \quad i \in I_-.$$

Assume first that $b \ge 0$. In this case, since $\boldsymbol{\alpha}^C$ is bounded, when C is sufficiently small, we will necessarily have $\lambda_i - \xi_i < 0$ for all $i \in I_+$. Pick such a C. Since ξ_i and λ_i are nonnegative, $\xi_i > 0$ for all $i \in I_+$, and from (9), $\alpha_i^C = \gamma/n$ for all $i \in I_+$. If $\gamma n_+/n \ge (1-\gamma)n_-/n$, then this $\boldsymbol{\alpha}^C$ is feasible and $\sum_{i=1}^n \alpha_i^C = \nu_{\max}$. However, if $\gamma n_+/n < (1-\gamma)n_-/n$, then we have a contradiction, and thus it must actually be that $b < 0$. In this case, for C sufficiently small, $\lambda_i - \xi_i < 0$ for all $i \in I_i$. As before, this now implies that $\alpha_i^C = (1-\gamma)/n$ for all $i \in I_-$, and thus $\sum_{i=1}^n \alpha_i^C = \nu_{\max}$. Hence, $\nu^* = \sum_{i=1}^n \alpha_i^C = \nu_{\max}$, and from Proposition 2, we immediately know that $(D_{2\nu})$ is infeasible if $\nu > \nu^*$.

For all $\nu \le \nu^*$, from Proposition 2, $(D_{2\nu})$ is feasible. From Lemma 4, we know that $\sum_{i=1}^n \alpha_i^C$ is a continuous decreasing function. Thus, for any $\nu \in (\nu_*, \nu^*]$, there is a $C > 0$ such that $\sum_{i=1}^n \alpha_i^C = \nu$, and by Lemma 2, any $\boldsymbol{\alpha}$ is an optimal solution of $(D_{2\nu})$ if and only if it is an optimal solution for (D'_{2C}).

Finally, we consider $\nu \in [0, \nu_*]$. If $\nu < \nu_*$, then $(D_{2\nu})$ must have an optimal objective value of zero because otherwise, by the definition of ν_*, this would contradict Theorem 2. If $\nu = \nu_* = 0$, then the optimal objective value of $(D_{2\nu})$ is zero, as $\boldsymbol{\alpha}^\nu = \mathbf{0}$ is a feasible solution. If $\nu = \nu_* > 0$, then Lemma 1 and the fact that the feasible region of $(D_{2\nu})$ is bounded by $0 \le \alpha_i \le \gamma/n$ for $i \in I_+$ and $0 \le \alpha_i \le (1-\gamma)/n$ for $i \in I_-$ imply that there exists a sequence $\{\boldsymbol{\alpha}^{\nu_j}\}$, $\nu_1 \le \nu_2 \le \cdots \le \nu_*$, such that $\boldsymbol{\alpha}^{\nu_j}$ is an optimal solution of $(D_{2\nu})$ with $\nu = \nu_j$, $\sum_{i=1}^n \alpha_i^{\nu_j} = \nu_j$, and $\boldsymbol{\alpha}^* = \lim_{\nu_j \to \nu_*} \boldsymbol{\alpha}^{\nu_j}$ exists. Since $\sum_{i=1}^n \alpha_i^{\nu_j} = \nu_j$,

$$\sum_{i=1}^n \alpha_i^* = \lim_{\nu_j \to \nu_*} \sum_{i=1}^n \alpha_i^{\nu_j} = \nu_*.$$

Since the feasible region of $(D_{2\nu})$ is a closed set, we also immediately have that $\boldsymbol{\alpha}^*$ is a feasible solution of $(D_{2\nu})$ for $\nu = \nu_*$. Since $\sum_{\ell,m=1}^n \alpha_\ell^{\nu_j} \alpha_m^{\nu_j} y_\ell y_m k(\mathbf{x}_\ell, \mathbf{x}_m) = 0$ for all ν_j, we find that $\sum_{\ell,m=1}^n \alpha_\ell^* \alpha_m^* y_\ell y_m k(\mathbf{x}_\ell, \mathbf{x}_m) = 0$ by taking the limit. Therefore, the optimal objective value of $(D_{2\nu})$ is zero if $\nu = \nu_*$. Thus, the optimal objective value of $(D_{2\nu})$ is zero for all $\nu \in [0, \nu_*]$.

Now suppose for the sake of a contradiction that the optimal objective value of $(D_{2\nu})$ is zero but $\nu > \nu_*$. By Lemma 4, there exists a $C > 0$ such that, if $\boldsymbol{\alpha}^C$ is an optimal solution of (D'_{2C}), then $\sum_{i=1}^n \alpha_i^C = \nu$. From Lemma 3, $\nu = \nu_{\max} = \nu^* = \nu_*$ since $\sum_{i=1}^n \alpha_i^C$ is the same for all C. This contradicts the assumption that $\nu > \nu_*$. Thus, the objective value of $(D_{2\nu})$ can be zero if and only if $\nu \le \nu_*$. In this case, $\mathbf{w} = 0$ and thus the solution is trivial.

By appropriate rescaling, this establishes the theorem. □

Acknowledgments

The work of Mark A. Davenport and Richard G. Baraniuk was supported by US National Science Foundation (NSF) Grant CCF-0431150 and the Texas Instruments Leadership University Program. The work of Clayton D. Scott was partially supported by NSF Vertical Integration of Research and Education grant 0240068 while he was a postdoctoral fellow at Rice University (for more details, see www.eecs.umich.edu/cscott).

References

[1] A. Cannon, J. Howse, D. Hush, and C. Scovel, “Learning with the Neyman-Pearson and Min-Max Criteria,” Technical Report LA-UR 02-2951, Los Alamos Nat’l Laboratory, 2002.

[2] F. Sebastiani, “Machine Learning in Automated Text Categorization,” *ACM Computing Surveys,* vol. 34, pp. 1-47, 2002.

[3] S. Bengio, J. Mariéthoz, and M. Keller, “The Expected Performance Curve,” *Proc. Int’l Conf. Machine Learning,* 2005.

[4] C.D. Scott and R.D. Nowak, “A Neyman-Pearson Approach to Statistical Learning,” *IEEE Trans. Information Theory,* vol. 51, no. 11, pp. 3806-3819, Nov. 2005.

[5] L.L. Scharf, *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis.* Addison-Wesley, 1991.

[6] H.G. Chew, R.E. Bogner, and C.C. Lim, “Dual-ν Support Vector Machine with Error Rate and Training Size Biasing,” *Proc. IEEE Int’l Conf. Acoustics, Speech, and Signal Processing,* pp. 1269-1272, 2001.

[7] E. Osuna, R. Freund, and F. Girosi, “Support Vector Machines: Training and Applications,” Technical Report A.I. Memo No. 1602, MIT Artificial Intelligence Laboratory, Mar. 1997.

[8] K. Veropoulos, N. Cristianini, and C. Campbell, “Controlling the Sensitivity of Support Vector Machines,” *Proc. Int’l Joint Conf. Artificial Intelligence,* 1999.

[9] Y. Lin, Y. Lee, and G. Wahba, “Support Vector Machines for Classification in Nonstandard Situations,” Technical Report No. 1016, Dept. of Statistics, Univ. of Wisconsin, Mar. 2000.

[10] M.A. Davenport, R.G. Baraniuk, and C.D. Scott, “Controlling False Alarms with Support Vector Machines,” *Proc. IEEE Int’l Conf. Acoustics, Speech, and Signal Processing,* 2006.

[11] M.A. Davenport, R.G. Baraniuk, and C.D. Scott, "Minimax Support Vector Machines," *Proc. IEEE Workshop Statistical Signal Processing,* 2007.

[12] M.A. Davenport, "Error Control for Support Vector Machines," MS thesis, Rice Univ., Apr. 2007.

[13] C.C. Chang and C.J. Lin, *LIBSVM: A Library for Support Vector Machines,* http://www.csie.ntu.edu.tw/cjlin/libsvm, 2001.

[14] B. Schölkopf and A.J. Smola, *Learning with Kernels.* MIT Press, 2002.

[15] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge Univ. Press, 2004.

[16] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning,* vol. 20, no. 3, pp. 273-297, 1995.

[17] B. Schölkopf, A.J. Smola, R. Williams, and P. Bartlett, "New Support Vector Algorithms," *Neural Computation,* vol. 12, pp. 1083-1121, 2000.

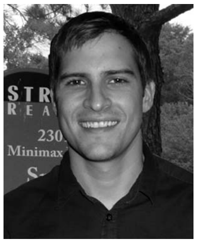
[18] C.C. Chang and C.J. Lin, "Training ν-Support Vector Classifiers: Theory and Algorithms," *Neural Computation,* vol. 13, pp. 2119-2147, 2001.

[19] C.D. Scott, "Performance Measures for Neyman-Pearson Classification," *IEEE Trans. Information Theory,* vol. 53, no. 8, pp. 2852-2863, Aug. 2007.

[20] J. Demšar, "Statistical Comparisons of Classifiers over Multiple Data Sets," *J. Machine Learning Research,* vol. 7, pp. 1-30, 2006.

[21] P.-H. Chen, C.-J. Lin, and B. Schölkopf, "A Tutorial on ν-Support Vector Machines," *Applied Stochastic Models in Business and Industry,* vol. 21, pp. 111-136, 2005.

[22] F. Bach, D. Heckerman, and E. Horvitz, "Considering Cost Asymmetry in Learning Classifiers," *J. Machine Learning Research,* vol. 7, pp. 1713-1741, 2006.



Mark A. Davenport received the BSEE, MS, and PhD degrees in electrical and computer engineering in 2004, 2007, and 2010, and the BA degree in managerial studies in 2004, all from Rice University, Houston, Texas. He is currently a US National Science Foundation Mathematical Sciences Postdoctoral Research Fellow in the Department of Statistics at Stanford University, Stanford, California. His research interests include compressive sensing, nonlinear approximation, and the application of low-dimensional signal models to a variety of problems in signal processing and machine learning. He is also cofounder and an editor of *Rejecta Mathematica*. Dr. Davenport shared the Hershel M. Rich Invention Award from Rice in 2007 for his work on the single-pixel camera and compressive sensing. He is a member of the IEEE.

Richard G. Baraniuk received the BSc degree in electrical engineering from the University of Manitoba, Canada, in 1987, the MSc degree in electrical engineering from the University of Wisconsin-Madison in 1988, and the PhD degree in electrical engineering from the University of Illinois at Urbana-Champaign in 1992. From 1992 to 1993, he was with the Signal Processing Laboratory of the Ecole Normale Supérieure in Lyon, France. Then, he joined Rice University, where he is currently the Victor E. Cameron Professor of Electrical and Computer Engineering. He spent sabbaticals at the Ecole Nationale Supérieure de Télécommunications in Paris in 2001 and the Ecole Fédérale Polytechnique de Lausanne in Switzerland in 2002. His research interests lie in the area of signal and image processing. He has been a guest editor of several special issues of the *IEEE Signal Processing Magazine,* the *IEEE Journal of Special Topics in Signal Processing,* and the *Proceedings of the IEEE*, and has served as technical program chair or on the technical program committee for several IEEE workshops and conferences. In 1999, he founded Connexions (cnx.org), a nonprofit publishing project that invites authors, educators, and learners worldwide to "create, rip, mix, and burn" free textbooks, courses, and learning materials from a global open-access repository. He received a NATO postdoctoral fellowship from NSERC in 1992, the National Young Investigator award from the US National Science Foundation in 1994, the Young Investigator Award from the US Office of Naval Research in 1995, the Rosenbaum Fellowship from the Isaac Newton Institute of Cambridge University in 1998, the C. Holmes MacDonald National Outstanding Teaching Award from Eta Kappa Nu in 1999, the Charles Duncan Junior Faculty Achievement Award from Rice in 2000, the University of Illinois ECE Young Alumni Achievement Award in 2000, the George R. Brown Award for Superior Teaching at Rice in 2001, 2003, and 2006, respectively, the Hershel M. Rich Invention Award from Rice in 2007, the Wavelet Pioneer Award from SPIE in 2008, and the Internet Pioneer Award from the Berkman Center for Internet and Society at Harvard Law School in 2008. He was selected as one of *Edutopia Magazine*'s Daring Dozen educators in 2007. Connexions received the Tech Museum Laureate Award from the Tech Museum of Innovation in 2006. Dr. Baraniuk's work with Kevin Kelly on the Rice single-pixel compressive camera was selected by *MIT Technology Review Magazine* as a TR10 Top 10 Emerging Technology in 2007. He was coauthor on a paper with Matthew Crouse and Robert Nowak that won the IEEE Signal Processing Society Junior Paper Award in 2001 and another with Vinay Ribeiro and Rolf Riedi that won the Passive and Active Measurement (PAM) Workshop Best Student Paper Award in 2003. He was elected a fellow of the IEEE in 2001 and a plus member of the AAA in 1986.

Clayton D. Scott received the AB degree in mathematics from Harvard University in 1998 and the MS and PhD degrees in electrical engineering from Rice University, in 2000 and 2004, respectively. He was a postdoctoral fellow in the Department of Statistics at Rice, and is currently an assistant professor in the Department of Electrical Engineering and Computer Science at the University of Michigan, Ann Arbor. His research interests include machine learning and statistical signal processing. He is a member of the IEEE.