Tuning Support Vector Machines for Minimax and Neyman-Pearson Classification

Mark A. Davenport, Richard G. Baraniuk, and Clayton D. Scott,

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.

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 is unknown. In such cases, two alternatives to expected misclassification 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

- *M. D. and R. G. are with the Department of Electrical and Computer Engineering, Rice University, Houston, TX, 77005. They are supported by NSF grant CCF-0431150 and the Texas Instruments Leadership University Program.*
  Email: {md, richb} – at – rice.edu, Web: dsp.rice.edu.
- *C. S. is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, 48109. He was partially supported by NSF Vertical Integration of Research and Education grant 0240068 while a postdoctoral fellow at Rice University.*
  Email: cscott – at – eecs.umich.edu, Web: www.eecs.umich.edu/~cscott.

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 $f : \mathbb{R}^d \to \{+1, -1\}$, let

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

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

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 classifiers operating at *equal error rate* or the *break even point*, where $P_F(f) = P_M(f)$ [1]–[3]. Of course, many classifiers may satisfy this constraint. We seek the best possible, the *minimax* classifier, which is defined as

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

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

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

It can be shown that if we consider

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

then both f^*_{MM} and f^*_α are equal to the solution of (5) for the appropriate values of γ. Thus, training an SVM for minimax and NP classification can be accomplished using a cost-sensitive SVM by tuning the parameter γ to achieve the desired error constraints. Observe that 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 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 pre-defined 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 [5]. We prove that this cost-sensitive SVM is equivalent to the more common $2C$-SVM [6]–[8] and provide a careful characterization of its parameter space in Section 2. We then leverage this characterization to develop a simple but powerful approach to error estimation based on smoothing cross-validation (CV) error estimates in Section 3, which also describes computationally efficient strategies for parameter selection. We conduct a detailed experimental evaluation in Sections 4 and 5 and demonstrate the superior performance of (*i*) our approaches to estimation relative to conventional CV and (*ii*) 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 [9]–[11]. Our software — based on the LIBSVM package [12] — 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 [13]. 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:

$$(P) \qquad \min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|^2$$

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

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 dual formulation obtained via the Karush-Kuhn-Tucker (KKT) conditions [14], where we substitute $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \Phi(\mathbf{x}_i)$ and solve for the optimal $\boldsymbol{\alpha} \in \mathbb{R}^n$. Note first that $\mathbf{w}$ depends only on the $\mathbf{x}_i$ for which $\alpha_i \neq 0$, which are called the *support vectors.* Second, 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. Hence, rather than choosing Φ directly, we can instead choose a *kernel* operator $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. If k is positive definite, then this defines a space $\mathcal{H}$ and a mapping Φ such that $k(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}')\rangle_{\mathcal{H}}$, which allows us to compute inner products in $\mathcal{H}$ without explicitly evaluating Φ.

To reduce sensitivity to outliers and allow for non-separable data, it is usually desirable to relax the constraint that each training vector is classified correctly through the introduction of *slack variables*, i.e., 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 adds a $\sum_{i=1}^{n} \xi_i$ penalty to the objective function. There are essentially two ways of doing this, which result in two different 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 [15]. An alternative (but equivalent) formulation is the ν-SVM [16]. The ν-SVM 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. The 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 [16].

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 [6]. Let $I_+ = \{i : y_i = +1\}$ and $I_- = \{i : y_i = -1\}$. The $2C$-SVM quadratic program has primal formulation

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

$$\begin{aligned} \text{s.t.} \qquad & y_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle_{\mathcal{H}} + b) \geq 1 - \xi_i && \text{for } i = 1, \ldots, n \\ & \xi_i \geq 0 && \text{for } i = 1, \ldots, n \end{aligned}$$

and dual formulation

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

$$\begin{aligned} \text{s.t.} \qquad & 0 \leq \alpha_i \leq C\gamma && \text{for } i \in I_+ \\ & 0 \leq \alpha_i \leq C(1-\gamma) && \text{for } i \in I_- \\ & \sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned}$$

where $\gamma \in [0, 1]$ controls the tradeoff 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 [6]–[8].

Similarly, [5] proposed the 2ν-SVM as a cost-sensitive extension of the ν-SVM. The 2ν-SVM has primal

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

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

and dual

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

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

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 $(D_{2\nu})$ is feasible if an only if $(\nu_+, \nu_-) \in [0,1]^2$; thus this parameterization lends itself naturally towards simple uniform grid searches and a number of additional heuristics 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 [5] concerning the interpretation of the parameters in the (ν_+, ν_-) formulation.

Proposition 1 ([5]): 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

$$ME_+ \leq \nu_+ \leq SV_+$$
$$ME_- \leq \nu_- \leq SV_-.$$

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

Proposition 2: Fix $\gamma \in [0, 1]$. Then $(D_{2\nu})$ is feasible if and only if $\nu \leq \nu_{\max} \leq 1$, where

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

Proof: First, assume that $\nu \leq \nu_{\max}$. Then there exists an $\boldsymbol{\alpha}$ that satisfies the constraints of $(D_{2\nu})$. Specifically, let

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

and

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

Then $\sum_{i \in I_+} \alpha_i + \sum_{i \in I_+} \alpha_i = \nu_{\max} \geq \nu$ and $\sum_{i=1}^n \alpha_i y_i = 0$. Thus $(D_{2\nu})$ is feasible.

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

Finally, we see that

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

as desired. □

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

Proposition 3: $(D_{2\nu})$ is feasible if and only if $\nu_+ \leq 1$ and $\nu_- \leq 1$.

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

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

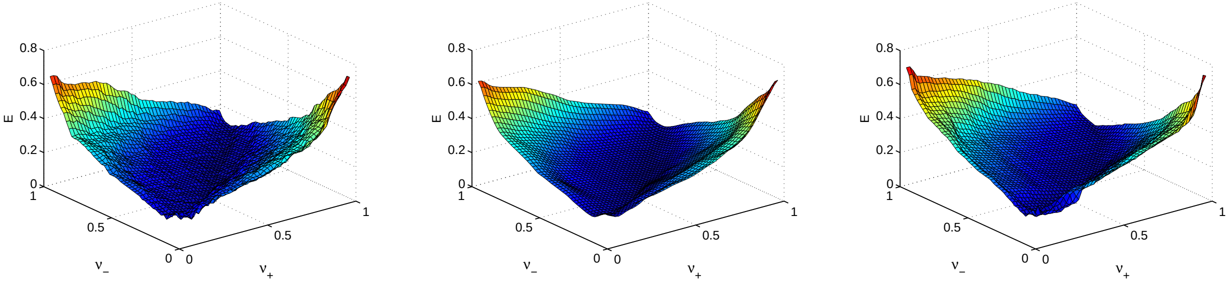
Thus, $(D_{2\nu})$ is feasible if and only if

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

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

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

The following theorems extend the results of [17] 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 towards more effective parameter selection procedures. The theorems



(a) (b) (c)

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

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 $(D_{2\nu})$.

Theorem 2: Fix $\gamma \in [0,1]$. For any $\nu \in (0, \nu_{\max}]$, assume $(D_{2\nu})$ has a nonzero optimal objective value, so that $\rho > 0$, and 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}). 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 $0 \le \nu_* \le \nu^* = \nu_{\max} \le 1$. 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 the 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 $C \ge C^*$. 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 method offers over the other is extremely important. In our case, the $2C$-SVM faces a number of problematic issues involved with an unbounded parameter space, such as numerical issues for very large or small parameter values and a certain degree of arbitrariness in selecting the starting points, ending points, and spacing for any grid search. 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_-). Thus we will restrict our attention to the 2ν-SVM. Our approach is to obtain estimates of the error rates over a grid of possible parameter values and then use these estimates to select the best parameter combination. The central focus of our study (which will be based on simulations across a wide range of data sets) concerns 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 3-D 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 $\widehat{P}_F^{\mathrm{CV}}$ and $\widehat{P}_M^{\mathrm{CV}}$. 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 $\widehat{E}^{\mathrm{CV}}$, where for minimax classification we set $\widehat{E}^{\mathrm{CV}} = \widehat{E}_{\mathrm{MM}}^{\mathrm{CV}} = \max\{\widehat{P}_F^{\mathrm{CV}}, \widehat{P}_M^{\mathrm{CV}}\}$ and for NP classification we set $\widehat{E}^{\mathrm{CV}} = \widehat{E}_{\mathrm{NP}(\alpha)}^{\mathrm{CV}}$ where $\widehat{E}_{\mathrm{NP}(\alpha)}^{\mathrm{CV}} = \widehat{P}_M^{\mathrm{CV}}$ when $\widehat{P}_F^{\mathrm{CV}} \leq \alpha$ and $\widehat{E}_{\mathrm{NP}(\alpha)}^{\mathrm{CV}} = \infty$ 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 datasets for the 2ν-SVM that $\widehat{P}_F^{\mathrm{CV}}$ and $\widehat{P}_M^{\mathrm{CV}}$ appear to vary smoothly as functions of (ν_+,ν_-,σ) but also appear to be somewhat "noisy" as illustrated in Fig. 1 (a). This motivates a simple heuristic to improve upon CV: smooth $\widehat{P}_F^{\mathrm{CV}}$ and $\widehat{P}_M^{\mathrm{CV}}$ with a low-pass filter W and then calculate $\widehat{E}^{\mathrm{SM}}$ 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 2-D 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 3-D filter to the error estimates, smoothing across different kernel parameter values. Fig. 1 illustrates the effect of 3-D smoothing on an example dataset, demonstrating that $\widehat{E}^{\mathrm{SM}}$ more closely resembles the estimate of E obtained from an independent test set. In our experiments, the filter is chosen to be a simple low-pass filter. Several possible filters can be used (for example, Gaussian filters, median filters, etc.), and all result in similar performance gains. The key to all these smoothing approaches is that they perform some kind of local averaging to reduce outlying estimates. We will see that both 2-D and 3-D methods are extremely effective in a quantitative sense in Section 5.

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 speed-up heuristic exists. Again inspired by the smoothness of $\widehat{P}_F^{\mathrm{CV}}$ and $\widehat{P}_M^{\mathrm{CV}}$, we propose a coordinate descent search. Several variants are possible, but the simplest one we employ, denoted 2-D coordinate descent, is described in Algorithm 2. It essentially consists of a sequence of orthogonal line searches that continues until it converges to a local optimum. To incorporate a kernel parameter, we can either repeat this approach for each value of the kernel parameter, or consider the natural 3-D extension of this algorithm. Smoothing can also easily be 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.

4 Experimental Setup

4.1 Performance evaluation

In order to evaluate the heuristics 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 datasets representing a variety of dimensions and sample sizes.[1] The datasets comprise a mixture of synthetic and real data. For each of the first 9 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 $\widehat{P}_F^{\mathrm{CV}}$ and $\widehat{P}_M^{\mathrm{CV}}$ for various parameter combinations using 5-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, and for NP classification we use the *Neyman-Pearson score*,

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

[1] We use the following datasets, 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.

Algorithm 1 Smoothed Grid Search

```
for a vector of values of $\nu_+$ do
  for a vector of values of $\nu_-$ do
    $\widehat{E}^{\mathrm{CV}} \leftarrow$ CV estimate of $E$
  end for
end for
$\widehat{E}^{\mathrm{SM}} \leftarrow W(\widehat{E}^{\mathrm{CV}})$
select $\nu_+$, $\nu_-$ minimizing $\widehat{E}^{\mathrm{SM}}$
train SVM using selected $\nu_+$, $\nu_-$
```

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 $\widehat{E}^{\mathrm{CV}}$
  increment $i$
until $\nu_+^i = \nu_+^{i-1}$ and $\nu_-^i = \nu_-^{i-1}$
train SVM using $\nu_+^i$, $\nu_-^i$
```

proposed in [18]. It can be shown that the global minimizer of (6) is the optimal NP classifier. 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 0, a stiffer penalty is exacted on classifiers that violate the constraint [18]. To evaluate performance on unbalanced datasets, we repeated these experiments retaining only 10% of the negatively labeled training data.

In order to compare multiple algorithms on multiple datasets, we use the two-step procedure advocated in [19]. 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 dataset, 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 [19] for a more thorough discussion of and motivation for these techniques.

4.2 Implementation

In all experiments we use a radial basis function (Gaussian) kernel and consider 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 2-

D 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 3-D 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.) We have implemented the 2ν-SVM (available online at www.dsp.rice.edu/software) by modifying the popular LIBSVM package [12].

4.3 Alternative approaches

In order to provide a reference for comparison, we also consider two alternative approaches, *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. Note that we do not expect that bias-shifting will perform as well as the 2ν-SVM since in [20] it was 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. In our experiments we search over a uniform grid of 50 points of the parameter ν and also apply a 3×3 smoothing filter to smooth the error estimates across different values of ν and σ.

A common motivation for minimax classification is that some datasets 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 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 datasets. 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 and variance of the two estimation approaches by averaging over different permutations. From Fig. 2, we see that smoothing leads to significant reductions in both bias and variance across all data sets. Notice also that the bias is always negative. 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 and variance reductions translate into a drastic improvement on the resulting classifiers. The results of smoothing on our benchmark datasets are shown in Table 1, and they clearly indicate that both 2-D and 3-D smoothing offer a statistically significant gain in performance, with 3-D smoothing offering a slight edge.

5.2 Coordinate descent

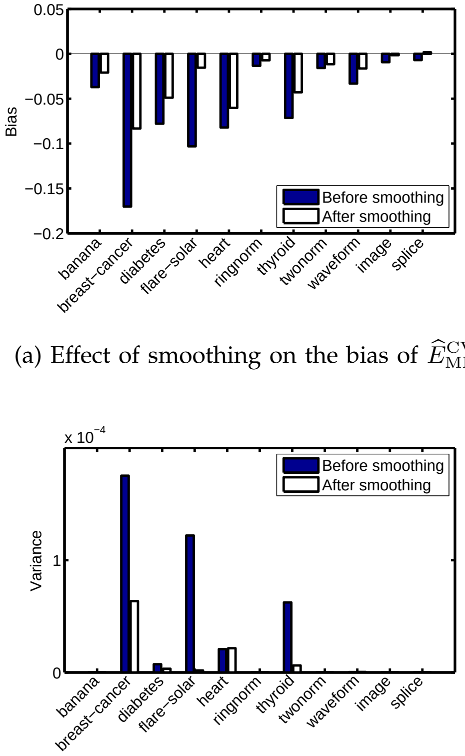
Table 2 shows that 3-D smoothing combined with either 2-D or 3-D coordinate descent offers gains in performance as well, which is particularly helpful since these methods speed up the parameter selection process considerably. Note that smoothing again makes a tremendous impact on the resulting performance, even in the absence of a complete grid search. Perhaps somewhat surprisingly, we observe that 2-D and 3-D coordinate descent behave similarly, despite 3-D 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 3-D smoothed grid-search approach (labeled 3D-SGS), the 2-D and 3-D coordinate descent methods (labeled 2D-CD and 3D-CD — both use 3-D smoothing), the balanced ν-SVM without bias-shifting (labeled Bal ν-SVM), and the traditional ν-SVM with bias-shifting (labeled ν-SVM). For the case of minimax classification on balanced balanced datasets, 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. Perhaps the most surprising result is that the 3-D coordinate descent method is not only competitive with the full grid search, but performs even better than the grid search on the unbalanced datasets. 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,



(a) Effect of smoothing on the bias of $\widehat{E}^{\mathrm{CV}}_{\mathrm{MM}}$.

(b) Effect of smoothing on the variance of $\widehat{E}^{\mathrm{CV}}_{\mathrm{MM}}$.

Fig. 2. *Effect of smoothing on* $\widehat{E}^{\mathrm{CV}}_{\mathrm{MM}}$.

TABLE 1

Average ranking of each smoothing approach for the 2ν-SVM. (Friedman p-values are < 0.01 for all cases; Nemenyi critical difference at 0.05 is 1.10.)

	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**

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 underlying 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 suprising

TABLE 2

Average ranking of each coordinate descent approach for the 2ν-SVM. (Friedman p-values are < 0.05 for all cases; Nemenyi critical difference at 0.05 is 1.92.)

	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**

TABLE 3

Average ranking of the 2ν-SVM methods, the balanced ν-SVM, and the ν-SVM with bias-shifting. (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.)

	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

performance of our greedy coordinate descent speed-up: By ignoring many parameter combinations, the algorithm reduces its exposure to such outliers.

Appendix

In [17] 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:

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

$$\text{s.t.} \qquad 0 \le \alpha_i \le \frac{\gamma}{n}, \qquad i \in I_+$$

$$0 \le \alpha_i \le \frac{1-\gamma}{n}, \qquad i \in I_-$$

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

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: This lemma was proved in [17] for the ν-SVM. The proof relies only on the form of the objective function of the dual formulation of the ν-SVM, which is identical to that of $(D_{2\nu})$. Thus, we omit it for the sake of brevity and refer the reader to [17]. □

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 proved in [17]. The proof depends only on the form of the objective functions, which are identical to those of (D'_{2C}) and $(D_{2\nu})$, and on the analogue of Lemma 1. Thus, we again refer the reader to [17]. □

For the proofs of Theorems 1 and 2, we will need to employ the Karush-Kuhn-Tucker (KKT) conditions [14]. 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:

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

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

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

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

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

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:

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

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

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

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

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

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 condition (12) we have

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

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

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

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

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

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 $\sum_{i=1}^n\sum_{j=1}^n\alpha_i^C\alpha_j^Cy_iy_jk(\mathbf{x}_i,\mathbf{x}_j)=0$. The kernel k is (by definition) positive definite, so we have $\sum_{j=1}^n\alpha_j^Cy_iy_jk(\mathbf{x}_i,\mathbf{x}_j)=0$. Thus, conditions (7) and (12) become

$$-\frac{1}{Cn}+by_i=\lambda_i-\xi_i\ \ \text{for}\ \ i=1,\dots,n,$$

or

$$\begin{aligned} -\frac{1}{Cn} + b &= \lambda_i - \xi_i \quad \text{for} \quad i \in I_+ \\ -\frac{1}{Cn} - b &= \lambda_i - \xi_i \quad \text{for} \quad i \in I_-. \end{aligned}$$

Assume first that $b \geq 0$; then

$$\lambda_i - \xi_i < 0 \quad \text{for} \quad i \in I_-.$$

This implies that $\xi_i > 0$ for all $i \in I_-$ since both λ_i and ξ_i are nonnegative. Therefore, in order for the first conditions of (9) and (14) 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 $\sum_{i \in I_+} \alpha_i^C = \sum_{i \in I_-} \alpha_i^C$. Therefore we need $\sum_{i \in I_+} \alpha_i^C = (1-\gamma)n_-/n \leq \gamma n_+/n$. Hence, if $(1-\gamma)n_- > \gamma n_+$, then we have reached a contradiction, and thus $b < 0$.

Therefore, assume without loss of generality that $b \geq 0$ (since we can always relabel the points so that this would be true), in which case $(1-\gamma)n_- \leq \gamma n_+$ and $\alpha_i^C = (1-\gamma)/n$ for all $i \in I_-$. There are three possibilities for $i \in I_+$:

1) $\lambda_i - \xi_i < 0$
2) $\lambda_i - \xi_i > 0$
3) $\lambda_i - \xi_i = 0$.

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 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 the conditions (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 $(1-\gamma)n_- \leq \gamma n_+$, 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 proved in [17]. 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 [17]. □

We are now ready to prove the main theorem:

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

For any optimal solution of (D'_{2C}), condition (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 $b \geq 0$. 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 condition (9), $\alpha_i^C = \gamma/n$ for all $i \in I_+$. If $\gamma n_+/n \geq (1-\gamma)n_-/n$, 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 $\nu \leq \nu^*$, 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 feasible region of $(D_{2\nu})$ is bounded by $0 \leq \alpha_i \leq \gamma/n$ for $i \in I_+$ and $0 \leq \alpha_i \leq (1-\gamma)/n$ for $i \in I_-$ imply that there exists a sequence $\{\boldsymbol{\alpha}^{\nu_j}\}$, $\nu_1 \leq \nu_2 \leq \cdots \leq \nu_*$ 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 $\sum_{l,m=1}^n \alpha_l^{\nu_j} \alpha_m^{\nu_j} y_l y_m k(\mathbf{x}_l, \mathbf{x}_m) = 0$ for all ν_j, we find that $\sum_{l,m=1}^n \alpha_l^* \alpha_m^* y_l y_m k(\mathbf{x}_l, \mathbf{x}_m) = 0$ 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 $\nu \leq \nu_*$. In this case, $\mathbf{w} = 0$ and thus the solution is trivial.

By appropriate rescaling, this establishes the theorem. □

References

[1] A. Cannon, J. Howse, D. Hush, and C. Scovel, "Learning with the Neyman-Pearson and min-max criteria," Tech. Rep. LA-UR 02-2951, Los Alamos National 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," in *Proc. Int. Conf. Machine Learning*, 2005, Bonn, Germany.

[4] C. D. Scott and R. D. Nowak, "A Neyman-Pearson approach to statistical learning," *IEEE Trans. Inform. Theory*, vol. 51, no. 11, pp. 3806–3819, 2005.

[5] H. G. Chew, R. E. Bogner, and C. C. Lim, "Dual-ν support vector machine with error rate and training size biasing," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing (ICASSP)*, 2001, pp. 1269–1272.

[6] E. Osuna, R. Freund, and F. Girosi, "Support vector machines: Training and applications," Tech. Rep. A.I. Memo No. 1602, MIT Artificial Intelligence Laboratory, March 1997.

[7] K. Veropoulos, N. Cristianini, and C. Campbell, "Controlling the sensitivity of support vector machines," in *Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI)*, 1999.

[8] Y. Lin, Y. Lee, and G. Wahba, "Support vector machines for classification in nonstandard situations," Tech. Rep. Technical Report No. 1016, University of Wisconsin, Dept. of Statistics, March, 2000.

[9] M. A. Davenport, R. G. Baraniuk, and C. D. Scott, "Controlling false alarms with support vector machines," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing (ICASSP)*, 2006, Toulouse, France.

[10] M. A. Davenport, R. G. Baraniuk, and C. D. Scott, "Minimax support vector machines," in *Proc. IEEE Work. Stat. Signal Processing (SSP)*, 2007, Madison, Wisconsin.

[11] M. A. Davenport, "Error control for support vector machines," M.S. thesis, Rice University, Houston, Texas, April 2007.

[12] C. C. Chang and C. J. Lin, *LIBSVM: a library for support vector machines*, 2001, See http://www.csie.ntu.edu.tw/~cjlin/libsvm.

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

[14] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, 2004.

[15] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[16] B. Schölkopf, A. J. Smola, R. Williams, and P. Bartlett, "New support vector algorithms," *Neural Computation*, vol. 12, pp. 1083–1121, 2000.

[17] C. C. Chang and C. J. Lin, "Training ν-support vector classifiers: Theory and algorithms," *Neural Computation*, vol. 13, pp. 2119–2147, 2001.

[18] C. D. Scott, "Performance measures for Neyman-Pearson classification," *IEEE Trans. Inform. Theory*, vol. 53, no. 8, pp. 2852–2863, 2007.

[19] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[20] F. Bach, D. Heckerman, and E. Horvitz, "Considering cost asymmetry in learning classifiers," *J. Machine Learning Research*, vol. 7, pp. 1713–1741, 2006.