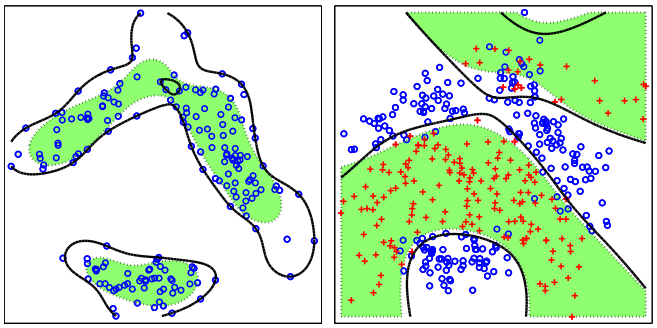
Nested Support Vector Machines

*Gyemin Lee, *Student Member, IEEE,* and Clayton Scott, *Member, IEEE*

***Abstract*—One-class and cost-sensitive support vector machines (SVMs) are state-of-the-art machine learning methods for estimating density level sets and solving weighted classification problems, respectively. However, the solutions of these SVMs do not necessarily produce set estimates that are nested as the parameters controlling the density level or cost-asymmetry are continuously varied. Such nesting not only reflects the true sets being estimated, but is also desirable for applications requiring the simultaneous estimation of multiple sets, including clustering, anomaly detection, and ranking. We propose new quadratic programs whose solutions give rise to nested versions of one-class and cost-sensitive SVMs. Furthermore, like conventional SVMs, the solution paths in our construction are piecewise linear in the control parameters, although here the number of breakpoints is directly controlled by the user. We also describe decomposition algorithms to solve the quadratic programs. These methods are compared to conventional (non-nested) SVMs on synthetic and benchmark data sets, and are shown to exhibit more stable rankings and decreased sensitivity to parameter settings.**

***Index Terms*—machine learning, pattern classification, one class support vector machine, cost sensitive support vector machine, nested set estimation, solution paths.**



(a) one-class SVM (b) cost-sensitive SVM

Fig. 1. Two decision boundaries from a one-class SVM (a) and a cost-sensitive SVM (b) at two density levels and cost asymmetries. The shaded regions indicate the density level set estimate at the higher density level and the positive decision set estimate at the lower cost asymmetry, respectively. These regions are not completely contained inside the solid contours corresponding to the smaller density level or the larger cost asymmetry, hence the two decision sets are not properly nested.

I. Introduction

Many statistical learning problems may be characterized as problems of *set estimation*. In these problems, the input takes the form of a random sample of points in a feature space, while the desired output is a subset G of the feature space. For example, in density level set estimation, a random sample from a density is given and G is an estimate of a density level set. In binary classification, labeled training data are available, and G is the set of all feature vectors predicted to belong to one of the classes.

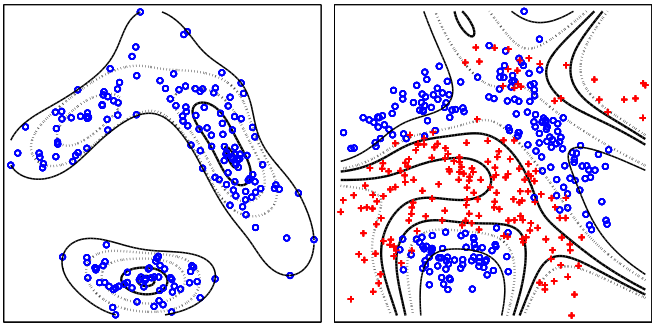
In other statistical learning problems, the desired output is a *family* of sets G_θ with the index θ taking values in a continuum. For example, estimating density level sets at multiple levels is an important task for many problems including clustering [1], outlier ranking [2], minimum volume set estimation [3], and anomaly detection [4]. Estimating cost-sensitive classifiers at a range of different cost asymmetries is important for ranking [5], Neyman-Pearson classification [6], semi-supervised novelty detection [7], and ROC studies [8].

Support vector machines (SVMs) are powerful nonparametric approaches to set estimation [9]. However, both the one-class SVM (OC-SVM) for level set estimation and the standard two-class SVM for classification do not produce set estimates that are *nested* as the parameter λ of the OC-SVM or, respectively, the misclassification cost of the two-class SVM is varied. As displayed in Fig. 1, set estimates from the original SVMs are not properly nested. On the other hand, Fig. 2 shows nested counterparts obtained from our proposed methods (see Section III, IV). Since the true sets being estimated are in fact nested, estimators that enforce the nesting constraint will not only avoid nonsensical solutions, but should also be more accurate and less sensitive to parameter settings and perturbations of the training data. One way to generate nested SVM classifiers is to train a cost-insensitive SVM and simply vary the offset. However, this often leads to inferior performance as demonstrated in [8].

Recently Clémençon and Vayatis [10] developed a method for bipartite ranking that also involves computing nested estimates of cost-sensitive classifiers at a finite grid of costs. Their set estimates are computed individually, and nesting is imposed subsequently through an explicit process of successive unions. These sets are then extended to a complete scoring function through piecewise constant interpolation. Their interest is primarily theoretical, as their estimates entail empirical risk minimization, and their results assume the underlying Bayes classifiers lies in a Vapnik-Chervonenkis class.

In this paper, we develop nested variants of one-class and two-class SVMs by incorporating nesting constraints into the dual quadratic programs associated with these methods. Decomposition algorithms for solving these modified duals are also presented. Like the solution paths for conventional SVMs [11], [8], [12], nested SVM solution paths are also piecewise linear in the control parameters, but require far fewer breakpoints. We compare our nested paths to the unnested paths on synthetic and benchmark data sets. We also quantify

G. Lee and C. Scott are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, 48109 USA. E-mail: {gyemin, cscott}@eecs.umich.edu. This work was supported in part by NSF Award No. 0830490.



(a) nested OC-SVM (b) nested CS-SVM

Fig. 2. Five decision boundaries from our nested OC-SVM (a) and nested CS-SVM (b) at five different density levels and cost asymmetries, respectively. These decision boundaries from nested SVMs do not cross each other, unlike the decision boundaries from the original SVMs (OC-SVM and CS-SVM). Therefore, the corresponding set estimates are properly nested.

the degree to which standard SVMs are unnested, which is often quite high. The Matlab implementation of our algorithms is available at http://www.eecs.umich.edu/~cscott/code/nestedsvm.zip. A preliminary version of this work appeared in [13].

A. Motivating Applications

With the multiple set estimates from nested SVMs over density levels or cost asymmetries, the following applications are envisioned.

Ranking : In the bipartite ranking problem [14], we are given labeled examples from two classes, and the goal is constructing a score function that rates new examples according to their likelihood of belonging to the positive class. If the decision sets are not nested as cost asymmetries or density levels varies, then the resulting score function leads to ambiguous ranking. Nested SVMs will make the ranking unambiguous and less sensitive to perturbations of the data. See Section VI-C for further discussion.

Clustering : Clusters may be defined as the connected components of a density level set. The level at which the density is thresholded determines a tradeoff between cluster number and cluster coverage. Varying the level from 0 to ∞ yields a "cluster tree" [15] that depicts the bifurcation of clusters into disjoint components and gives a hierarchical representation of cluster structure.

Anomaly Detection : Anomaly detection aims to identify deviations from nominal data when combined observations of nominal and anomalous data are given. Scott and Kolaczyk [4] and Scott and Blanchard [7] present approaches to classifying the contaminated, unlabeled data by solving multiple level set estimation and multiple cost-sensitive classification problems, respectively.

II. Background on CS-SVM and OC-SVM

In this section, we will overview two SVM variants and show how they can be used to learn set estimates. To establish notation and basic concepts, we briefly review SVMs.

Suppose that we have a random sample $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ where $\mathbf{x}_i \in \mathbb{R}^d$ is a feature vector and $y_i \in \{-1, +1\}$ is its class. An SVM finds a separating hyperplane with a normal vector $\mathbf{w}$ in a high dimensional space $\mathcal{H}$ by solving

$$\min_{\mathbf{w},\boldsymbol{\xi}} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \sum_i \xi_i$$
$$\text{s.t.} \quad y_i\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i$$

where λ is a regularization parameter and Φ is a nonlinear function that maps each data point into $\mathcal{H}$ generated by a positive semi-definite kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. This kernel corresponds to an inner product in $\mathcal{H}$ through $k(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}')\rangle$. Then the two half-spaces of the hyperplane $\{\Phi(\mathbf{x}) : f(\mathbf{x}) \equiv \langle \mathbf{w}, \Phi(\mathbf{x})\rangle = 0\}$ form positive and negative decision sets. Since the offset of the hyperplane is often omitted when Gaussian or inhomogeneous polynomial kernels are chosen [16], it is not considered in this formulation. More detailed discussion on SVMs can be found in [9].

A. Cost-Sensitive SVM

The SVM above, which we call a cost-insensitive SVM (CI-SVM), penalizes errors in both classes equally. However, there are many applications where the numbers of data samples from each class are not balanced, or false positives and false negatives incur different costs. The cost-sensitive SVM (CS-SVM) handles this issue by controlling the cost asymmetry between false positives and false negatives [17].

Let $I_+ = \{i : y_i = +1\}$ and $I_- = \{i : y_i = -1\}$ denote the two index sets, and γ denote the cost asymmetry. Then a CS-SVM solves

$$\min_{\mathbf{w},\boldsymbol{\xi}} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \gamma \sum_{I_+} \xi_i + (1-\gamma)\sum_{I_-} \xi_i \tag{1}$$
$$\text{s.t.} \quad y_i\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i$$

where $\mathbf{w}$ is the normal vector of the hyperplane. When $\gamma = \frac{1}{2}$, CS-SVMs reduce to CI-SVMs.

In practice this optimization problem is solved via its dual, which depends only on a set of Lagrange multipliers (one for each $\mathbf{x}_i$):

$$\min_{\boldsymbol{\alpha}} \frac{1}{2\lambda}\sum_i\sum_j \alpha_i\alpha_j y_i y_j K_{i,j} - \sum_i \alpha_i \tag{2}$$
$$\text{s.t.} \quad 0 \leq \alpha_i \leq \mathbf{1}_{\{y_i<0\}} + y_i\gamma, \quad \forall i.$$

where $K_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$ and $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_N)$. The indicator function $\mathbf{1}_{\{A\}}$ returns 1 if the condition A is true and 0 otherwise. Since there is no offset term, a linear constraint $\sum_i \alpha_i y_i = 0$ does not appear in the dual.

Once an optimal solution $\boldsymbol{\alpha}^*(\gamma) = (\alpha_1^*(\gamma), \ldots, \alpha_N^*(\gamma))$ is found, the sign of the decision function

$$f_\gamma(\mathbf{x}) = \frac{1}{\lambda}\sum_i \alpha_i^*(\gamma) y_i k(\mathbf{x}, \mathbf{x}_i) \tag{3}$$

determines the class of $\mathbf{x}$. If $k(\cdot,\cdot) \geq 0$, then this decision function takes only non-positive values when $\gamma = 0$, and corresponds to $(0, 0)$ in the ROC. On the other hand, $\gamma = 1$

penalizes only the violations of positive examples, and corresponds to $(1, 1)$ in the ROC.

Bach et al. [8] extended the method of Hastie et al. [11] to the CS-SVM. They showed that $\alpha_i^*(\gamma)$ are piecewise linear in γ, and derived an efficient algorithm for computing the entire path of solutions to (2). Thus, a family of classifiers at a range of cost asymmetries can be found with a computational cost comparable to solving (2) for a single γ.

B. One-Class SVM

The OC-SVM was proposed in [18], [19] to estimate a level set of an underlying probability density given a data sample from the density. In one-class problems, all the instances are assumed from the same class. The primal quadratic program of the OC-SVM is

$$\min_{\mathbf{w},\boldsymbol{\xi}} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{N}\sum_{i=1}^{N}\xi_i \quad (4)$$
$$\text{s.t.} \quad \langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i.$$

This problem is again solved via its dual in practice:

$$\min_{\boldsymbol{\alpha}} \frac{1}{2\lambda}\sum_i\sum_j \alpha_i\alpha_j K_{i,j} - \sum_i \alpha_i \quad (5)$$
$$\text{s.t.} \quad 0 \leq \alpha_i \leq \frac{1}{N}, \quad \forall i.$$

This formulation is equivalent to the more common ν parametrization [18], and is more convenient for our purposes. We also note that the OC-SVM can be solved by setting $\gamma = 1/2$ and $y_i = 1$ in the CS-SVM. However, our path algorithm for the OC-SVM, which varies λ, is not a special case of our path algorithm for the CS-SVM, which varies γ while holding λ fixed.

A solution $\boldsymbol{\alpha}^*(\lambda) = (\alpha_1^*(\lambda), \ldots, \alpha_N^*(\lambda))$ defines a decision function that determines whether a point is an outlier or not. Here $\alpha_i^*(\lambda)$ are also piecewise linear in λ [12]. From this property, we can develop a path following algorithm and generate a family of level set estimates with a small computational cost. The set estimate conventionally associated with the OC-SVM is given by

$$\widehat{G}_\lambda = \{\mathbf{x} : \sum_i \alpha_i^*(\lambda) k(\mathbf{x}_i, \mathbf{x}) > \lambda\}. \quad (6)$$

Vert and Vert [20] showed that by modifying this estimate slightly, substituting $\alpha_i^*(\eta\lambda)$ for $\alpha_i^*(\lambda)$ where $\eta > 1$, (6) leads to a consistent estimate of the true level set when a Gaussian kernel with a well-calibrated bandwidth is used. Regardless of whether $\eta = 1$ or $\eta > 1$, however, the obtained estimates are not guaranteed to be nested as we will see in Section VI. Note also that when $\alpha_i^*(\lambda) = \frac{1}{N}$, (6) is equivalent to set estimation based on kernel density estimation.

III. Nested CS-SVM

In this section, we develop the nested cost-sensitive SVM (NCS-SVM), which aims to produce nested positive decision sets $G_\gamma = \{\mathbf{x} : f_\gamma(\mathbf{x}) > 0\}$ as the cost asymmetry γ varies. Our construction is a two stage process. We first select a finite number of cost asymmetries $0 = \gamma_1 < \gamma_2 < \ldots < \gamma_M = 1$ a priori and generate a family of nested decision sets at the preselected cost asymmetries. We achieve this goal by incorporating nesting constraints into the dual quadratic program of CS-SVM. Second, we linearly interpolate the solution coefficients of the finite nested collection to a continuous nested family defined for all γ. As an efficient method to solve the formulated problem, we present a decomposition algorithm.

A. Finite Family of Nested Sets

Our NCS-SVM finds decision functions at cost asymmetries $\gamma_1, \gamma_2, \ldots, \gamma_M$ simultaneously by minimizing the sum of duals (2) at each γ and by imposing additional constraints that induce nested sets. For a fixed λ and preselected cost asymmetries $0 = \gamma_1 < \gamma_2 < \cdots < \gamma_M = 1$, an NCS-SVM solves

$$\min_{\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_M} \sum_{m=1}^{M}\left[\frac{1}{2\lambda}\sum_{i,j}\alpha_{i,m}\alpha_{j,m}y_iy_jK_{i,j} - \sum_i \alpha_{i,m}\right] \quad (7)$$
$$\text{s.t.} \quad 0 \leq \alpha_{i,m} \leq \mathbf{1}_{\{y_i<0\}} + y_i\gamma_m, \quad \forall i, m \quad (8)$$
$$y_i\alpha_{i,1} \leq y_i\alpha_{i,2} \leq \cdots \leq y_i\alpha_{i,M}, \quad \forall i \quad (9)$$

where $\boldsymbol{\alpha}_m = (\alpha_{1,m}, \ldots, \alpha_{N,m})$ and $\alpha_{i,m}$ is a coefficient for data point $\mathbf{x}_i$ and cost asymmetry γ_m. Then its optimal solution $\boldsymbol{\alpha}_m^* = (\alpha_{1,m}^*, \ldots, \alpha_{N,m}^*)$ defines the decision function $f_{\gamma_m}(\mathbf{x}) = \frac{1}{\lambda}\sum_i \alpha_{i,m}^* y_i k(\mathbf{x}_i, \mathbf{x})$ and its corresponding decision set $\widehat{G}_{\gamma_m} = \{\mathbf{x} : f_{\gamma_m}(\mathbf{x}) > 0)\}$ for each m. In Section VII, the proposed quadratic program for NCS-SVMs is interpreted as a dual of a corresponding primal quadratic program.

B. Interpolation

For an intermediate cost asymmetry γ between two cost asymmetries, say γ_1 and γ_2 without loss of generality, we can write $\gamma = \epsilon\gamma_1 + (1-\epsilon)\gamma_2$ for some $\epsilon \in [0, 1]$. Then we define new coefficients $\alpha_i^*(\gamma)$ through linear interpolation:

$$\alpha_i^*(\gamma) = \epsilon\alpha_{i,1}^* + (1-\epsilon)\alpha_{i,2}^*. \quad (10)$$

Then the positive decision set at cost asymmetry γ is

$$\widehat{G}_\gamma = \{\mathbf{x} : f_\gamma(\mathbf{x}) = \frac{1}{\lambda}\sum_i \alpha_i^*(\gamma) y_i k(\mathbf{x}_i, \mathbf{x}) > 0\}. \quad (11)$$

This is motivated by the piecewise linearity of the Lagrange multipliers of the CS-SVM, and is further justified by the following result.

Proposition 1. *The nested CS-SVM equipped with a kernel such that $k(\cdot,\cdot) \geq 0$ (e.g., Gaussian kernels or polynomial kernels of even orders) generates nested decision sets. In other words, if $0 \leq \gamma_\epsilon < \gamma_\delta \leq 1$, then $\widehat{G}_{\gamma_\epsilon} \subset \widehat{G}_{\gamma_\delta}$.*

Proof: We prove the proposition in three steps. First, we show that sets from (7) satisfy $\widehat{G}_{\gamma_1} \subset \widehat{G}_{\gamma_2} \subset \cdots \subset \widehat{G}_{\gamma_M}$. Second, we show that if $\gamma_m < \gamma < \gamma_{m+1}$, then $\widehat{G}_{\gamma_m} \subset \widehat{G}_\gamma \subset \widehat{G}_{\gamma_{m+1}}$. Finally, we prove that any two sets from the NCS-SVM are nested.

Without loss of generality, we show $\widehat{G}_{\gamma_1} \subset \widehat{G}_{\gamma_2}$. Let $\boldsymbol{\alpha}_1^*$ and $\boldsymbol{\alpha}_2^*$ denote the optimal solutions for γ_1 and γ_2. Then from $k(\cdot,\cdot) \geq 0$ and (9), we have $\sum_i \alpha_{i,1}^* y_i k(\mathbf{x}_i, \mathbf{x}) \leq \sum_i \alpha_{i,2}^* y_i k(\mathbf{x}_i, \mathbf{x})$. Therefore, $\widehat{G}_{\gamma_1} = \{\mathbf{x} : f_{\gamma_1}(\mathbf{x}) > 0\} \subset \widehat{G}_{\gamma_2} = \{\mathbf{x} : f_{\gamma_2}(\mathbf{x}) > 0\}$.

Next, without loss of generality, we show $\widehat{G}_{\gamma_1} \subset \widehat{G}_{\gamma} \subset \widehat{G}_{\gamma_2}$ when $\gamma_1 \leq \gamma \leq \gamma_2$. The linear interpolation (10) and the nesting constraints (9) imply $y_i \alpha_{i,1}^* \leq y_i \alpha_i^*(\gamma) \leq y_i \alpha_{i,2}^*$, which, in turn, leads to $\sum_i \alpha_{i,1}^* y_i k(\mathbf{x}_i, \mathbf{x}) \leq \sum_i \alpha_i^*(\gamma) y_i k(\mathbf{x}_i, \mathbf{x}) \leq \sum_i \alpha_{i,2}^* y_i k(\mathbf{x}_i, \mathbf{x})$.

Now consider arbitrary $0 \leqq \gamma_\epsilon < \gamma_\delta \leq 1$. If $\gamma_\epsilon \leq \gamma_m \leq \gamma_\delta$ for some m, then $\widehat{G}_{\gamma_\epsilon} \subset \widehat{G}_{\gamma_\delta}$ by the above results. Thus, suppose this is not the case and assume $\gamma_1 < \gamma_\epsilon < \gamma_\delta < \gamma_2$ without loss of generality. Then there exist $\epsilon > \delta$ such that $\gamma_\epsilon = \epsilon\gamma_1 + (1-\epsilon)\gamma_2$ and $\gamma_\delta = \delta\gamma_1 + (1-\delta)\gamma_2$. Suppose $\mathbf{x} \in \widehat{G}_{\gamma_\epsilon}$. Then $\mathbf{x} \in \widehat{G}_{\gamma_2}$, hence $f_{\gamma_\epsilon}(\mathbf{x}) = \frac{1}{\lambda}\sum_i (\epsilon\alpha_{i,1}^* + (1-\epsilon)\alpha_{i,2}^*) y_i k(\mathbf{x}_i, \mathbf{x}) > 0$ and $f_{\gamma_2}(\mathbf{x}) = \frac{1}{\lambda}\sum_i \alpha_{i,2}^* y_i k(\mathbf{x}_i, \mathbf{x}) > 0$. By adding $\frac{\delta}{\epsilon} f_{\gamma_\epsilon}(\mathbf{x}) + (1 - \frac{\delta}{\epsilon}) f_{\gamma_2}(\mathbf{x})$, we have $f_{\gamma_\delta}(\mathbf{x}) = \sum_i (\delta\alpha_{i,1}^* + (1-\delta)\alpha_{i,2}^*) y_i k(\mathbf{x}_i, \mathbf{x}) > 0$. Thus, $\widehat{G}_{\gamma_\epsilon} \subset \widehat{G}_{\gamma_\delta}$. ■

The assumption that the kernel is positive can in some cases be attained through pre-processing of the data. For example, a cubic polynomial kernel can be applied if the data support is shifted to lie in the positive orthant, so that the kernel function is in fact always positive.

C. Decomposition Algorithm

The objective function (7) requires optimization over $N \times M$ variables. Due to its large size, standard quadratic programming algorithms are inadequate. Thus, we develop a decomposition algorithm that iteratively divides the large optimization problem into subproblems and optimizes the smaller problems. A similar approach also appears in a multi-class classification algorithm [21], although the algorithm developed there is substantively different from ours. The decomposition algorithm follows:

1) Choose an example $\mathbf{x}_i$ from the data set.
2) Optimize coefficients $\{\alpha_{i,m}\}_{m=1}^M$ corresponding to $\mathbf{x}_i$ while leaving other variables fixed.
3) Repeat 1 and 2 until the optimality condition error falls below a predetermined tolerance.

The pseudo code given in Fig. 3 initializes with a feasible solution $\alpha_{i,m} = \mathbf{1}_{\{y_i<0\}} + y_i\gamma_m,\ \forall i, m$. A simple way of selection and termination is cycling through all the $\mathbf{x}_i$ or picking $\mathbf{x}_i$ randomly and stopping after a fixed number of iterations. However, by checking the Karush-Kuhn-Tucker (KKT) optimality conditions and choosing $\mathbf{x}_i$ most violating the conditions [22], the algorithm will converge in far fewer iterations. In the Appendix, we provide a detailed discussion of the data point selection scheme and termination criterion based on the KKT optimality conditions.

In step 2, the algorithm optimizes a set of variables associated to the chosen data point. Without loss of generality, let us assume that the data point $\mathbf{x}_1$ is chosen and $\{\alpha_{1,m}\}_{m=1}^M$ will be optimized while fixing the other $\alpha_{i,m}$. We rewrite the objective function (7) in terms of $\alpha_{1,m}$:

Input: $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, $\{\gamma_m\}_{m=1}^M$

Initialize:

$$\alpha_{i,m} \leftarrow \mathbf{1}_{\{y_i<0\}} + y_i\gamma_m, \quad \forall i, m$$

repeat

Choose a data point $\mathbf{x}_i$.

Compute:

$$f_{i,m} \leftarrow \frac{1}{\lambda}\sum_j \alpha_{j,m} y_j K_{i,j}, \quad \forall m$$

$$\alpha_{i,m}^{\text{new}} \leftarrow \alpha_{i,m} + \frac{\lambda(1 - y_i f_{i,m})}{K_{i,i}}, \quad \forall m$$

Update $\{\alpha_{i,m}\}_{m=1}^M$ with the solution of the subproblem:

$$\min_{\alpha_{i,1},\ldots,\alpha_{i,M}} \sum_m \left[\frac{1}{2}\alpha_{i,m}^2 - \alpha_{i,m}\alpha_{i,m}^{\text{new}}\right]$$
$$\text{s.t.} \quad 0 \leq \alpha_{i,m} \leq \mathbf{1}_{\{y_i<0\}} + y_i\gamma_m, \quad \forall m$$
$$y_i\alpha_{i,1} \leq y_i\alpha_{i,2} \leq \cdots \leq y_i\alpha_{i,M}$$

until Accuracy conditions are satisfied

Output: $\widehat{G}_{\gamma_m} = \{\mathbf{x} : \sum_i \alpha_{i,m} y_i k(\mathbf{x}_i, \mathbf{x}) > 0\}, \quad \forall m$

Fig. 3. Decomposition algorithm for a nested cost-sensitive SVM. Specific strategies for data point selection and termination, based on the KKT conditions, are given in the Appendix.

$$\begin{aligned}
&\sum_m \left[\frac{1}{2\lambda}\sum_{i,j}\alpha_{i,m}\alpha_{j,m}y_iy_jK_{i,j} - \sum_i \alpha_{i,m}\right] \\
=&\frac{1}{\lambda}\sum_m \left[\frac{1}{2}\alpha_{1,m}^2 K_{1,1} + \alpha_{1,m}\left(\sum_{j\neq 1}\alpha_{j,m}y_1y_jK_{1,j} - \lambda\right)\right] + C \\
=&\frac{1}{\lambda}\sum_m \left[\frac{1}{2}\alpha_{1,m}^2 K_{1,1} + \alpha_{1,m}\left(\lambda y_1 f_{1,m} - \alpha_{1,m}^{\text{old}}K_{1,1} - \lambda\right)\right] + C \\
=&\frac{K_{1,1}}{\lambda}\sum_m \left[\frac{1}{2}\alpha_{1,m}^2 - \alpha_{1,m}\left(\alpha_{1,m}^{\text{old}} + \frac{\lambda(1-y_1f_{1,m})}{K_{1,1}}\right)\right] + C
\end{aligned}$$

where $f_{1,m} = \frac{1}{\lambda}\left(\sum_{j\neq 1}\alpha_{j,m}y_jK_{1,j} + \alpha_{1,m}^{\text{old}}y_1K_{1,1}\right)$ and $\alpha_{1,m}^{\text{old}}$ denote the output and the variable preceding the update. These values can be easily computed from the previous iteration result. C is a collection of terms that do not depend on $\alpha_{1,m}$.

Then the algorithm solves the new subproblem with M variables,

$$\min_{\alpha_{1,1},\ldots,\alpha_{1,M}} \sum_m \left[\frac{1}{2}\alpha_{1,m}^2 - \alpha_{1,m}\alpha_{1,m}^{\text{new}}\right]$$
$$\text{s.t.} \quad 0 \leq \alpha_{1,m} \leq \mathbf{1}_{\{y_1<0\}} + y_1\gamma_m, \quad \forall m$$
$$y_1\alpha_{1,1} \leq y_1\alpha_{1,2} \leq \cdots \leq y_1\alpha_{1,M}$$

where $\alpha_{1,m}^{\text{new}} = \alpha_{1,m}^{\text{old}} + \frac{\lambda(1-y_1f_{1,m})}{K_{1,1}}$ is the solution if feasible. This subproblem is much smaller and can be solved efficiently via standard quadratic program solvers.

IV. Nested OC-SVM

In this section, we present a nested extension of OC-SVM. The nested OC-SVM (NOC-SVM) estimates a family

of nested level sets over a continuum of levels λ. Our approach here parallels the approach developed for the NCS-SVM. First, we will introduce an objective function for nested set estimation, and will develop analogous interpolation and decomposition algorithms for the NOC-SVM.

A. Finite Family of Nested Sets

For M different density levels of interest $\lambda_1 > \lambda_2 > \cdots > \lambda_M > 0$, an NOC-SVM solves the following optimization problem

$$\min_{\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_M} \sum_{m=1}^{M} \left[\frac{1}{2\lambda_m} \sum_{i,j} \alpha_{i,m}\alpha_{j,m}K_{i,j} - \sum_i \alpha_{i,m} \right] \quad (12)$$

$$\text{s.t.} \quad 0 \le \alpha_{i,m} \le \frac{1}{N}, \quad \forall i,m \quad (13)$$

$$\frac{\alpha_{i,1}}{\lambda_1} \le \frac{\alpha_{i,2}}{\lambda_2} \le \cdots \le \frac{\alpha_{i,M}}{\lambda_M}, \quad \forall i \quad (14)$$

where $\boldsymbol{\alpha}_m = (\alpha_{1,m},\ldots,\alpha_{N,m})$ and $\alpha_{i,m}$ corresponds to data point $\mathbf{x}_i$ at level λ_m. Its optimal solution $\boldsymbol{\alpha}_m^* = (\alpha_{1,m}^*,\ldots,\alpha_{N,m}^*)$ determines a level set estimate $\widehat{G}_{\lambda_m} = \{\mathbf{x} : f_{\lambda_m}(\mathbf{x}) > 1\}$ where $f_{\lambda_m}(\mathbf{x}) = \frac{1}{\lambda_m}\sum_i \alpha_{i,m}^* k(\mathbf{x}_i,\mathbf{x})$. In practice, we can choose λ_1 and λ_M to cover the entire range of interesting values of density level (see Section VI-B, Appendix C). In Section VII, this quadratic program for the NOC-SVM is interpreted as a dual of a corresponding primal quadratic program.

B. Interpolation and Extrapolation

We construct a density level set estimate at an intermediate level λ between two preselected levels, say λ_1 and λ_2. At $\lambda = \epsilon\lambda_1 + (1-\epsilon)\lambda_2$ for some $\epsilon \in [0,1]$, we set

$$\alpha_i^*(\lambda) = \epsilon\alpha_{i,1}^* + (1-\epsilon)\alpha_{i,2}^*.$$

For $\lambda > \lambda_1$, we extrapolate the solution by setting $\alpha_i^*(\lambda) = \alpha_{i,1}^*$ for $\forall i$. These are motivated by the facts that the OC-SVM solution is piecewise linear in λ and remains constant for $\lambda > \lambda_1$ as presented in Appendix C. Then the level set estimate becomes

$$\widehat{G}_\lambda = \{\mathbf{x} : \sum_i \alpha_i^*(\lambda)k(\mathbf{x}_i,\mathbf{x}) > \lambda\}. \quad (15)$$

The level set estimates generated from the above process are shown to be nested in the next Proposition.

Proposition 2. *The nested OC-SVM equipped with a kernel such that $k(\cdot,\cdot) \ge 0$ (in particular, a Gaussian kernel) generates nested density level set estimates. That is, if $0 < \lambda_\epsilon < \lambda_\delta < \infty$, then $\widehat{G}_{\lambda_\epsilon} \supset \widehat{G}_{\lambda_\delta}$.*

Proof: We prove the proposition in three steps. First, we show that sets from (12) satisfy $\widehat{G}_{\lambda_1} \subset \widehat{G}_{\lambda_2} \subset \cdots \subset \widehat{G}_{\lambda_M}$. Second, the interpolated set (15) is shown to satisfy $\widehat{G}_{\lambda_m} \subset \widehat{G}_\lambda \subset \widehat{G}_{\lambda_{m+1}}$ when $\lambda_m > \lambda > \lambda_{m+1}$. Finally, we prove the claim for any two sets from the NOC-SVM.

Without loss of generality, we first show $\widehat{G}_{\lambda_1} \subset \widehat{G}_{\lambda_2}$. Let $\lambda_1 > \lambda_2$ denote two density levels chosen a priori, and $\boldsymbol{\alpha}_1^*$ and $\boldsymbol{\alpha}_2^*$ denote their corresponding optimal solutions. From (14), we have $\sum_i \frac{\alpha_{i,1}^*}{\lambda_1}k(\mathbf{x}_i,\mathbf{x}) \le \sum_i \frac{\alpha_{i,2}^*}{\lambda_2}k(\mathbf{x}_i,\mathbf{x})$, so the two estimated level sets are nested $\widehat{G}_{\lambda_1} \subset \widehat{G}_{\lambda_2}$.

Next, without loss of generality, we prove $\widehat{G}_{\lambda_1} \subset \widehat{G}_\lambda \subset \widehat{G}_{\lambda_2}$ for $\lambda_1 > \lambda > \lambda_2$. From (14), we have $\frac{\alpha_{i,1}^*}{\lambda_1} \le \frac{\alpha_{i,2}^*}{\lambda_2}$ and

$$\begin{aligned}\frac{\alpha_{i,1}^*}{\lambda_1} &= \frac{\lambda\frac{\alpha_{i,1}^*}{\lambda_1}}{\lambda} = \frac{\epsilon\alpha_{i,1}^* + (1-\epsilon)\frac{\lambda_2}{\lambda_1}\alpha_{i,1}^*}{\lambda} \\ &\le \frac{\epsilon\alpha_{i,1}^* + (1-\epsilon)\alpha_{i,2}^*}{\lambda} = \frac{\alpha_i^*(\lambda)}{\lambda} \\ &\le \frac{\epsilon\frac{\lambda_1}{\lambda_2}\alpha_{i,2}^* + (1-\epsilon)\alpha_{i,2}^*}{\lambda} = \frac{\lambda\frac{\alpha_{i,2}^*}{\lambda_2}}{\lambda} = \frac{\alpha_{i,2}^*}{\lambda_2}.\end{aligned}$$

Hence, $f_{\lambda_1}(\mathbf{x}) \le f_\lambda(\mathbf{x}) \le f_{\lambda_2}(\mathbf{x})$.

Now consider arbitrary $\lambda_\delta > \lambda_\epsilon > 0$. By construction, we can easily see that $\widehat{G}_{\lambda_\delta} \subset \widehat{G}_{\lambda_\epsilon} \subset \widehat{G}_{\lambda_1}$ for $\lambda_\delta > \lambda_\epsilon > \lambda_1$, and $\widehat{G}_{\lambda_M} \subset \widehat{G}_{\lambda_\delta} \subset \widehat{G}_{\lambda_\epsilon}$ for $\lambda_M > \lambda_\delta > \lambda_\epsilon$. Thus we only need to consider the case $\lambda_1 > \lambda_\delta > \lambda_\epsilon > \lambda_M$. Since above results imply $\widehat{G}_{\lambda_\delta} \subset \widehat{G}_{\lambda_\epsilon}$ if $\lambda_\delta > \lambda_m > \lambda_\epsilon$ for some m, we can safely assume $\lambda_1 > \lambda_\delta > \lambda_\epsilon > \lambda_2$ without loss of generality. Then there exist $\delta > \epsilon$ such that $\lambda_\delta = \delta\lambda_1 + (1-\delta)\lambda_2$ and $\lambda_\epsilon = \epsilon\lambda_1 + (1-\epsilon)\lambda_2$. Suppose $\mathbf{x} \in \widehat{G}_{\lambda_\delta}$. Then $\mathbf{x} \in \widehat{G}_{\lambda_2}$ and

$$\sum_i (\delta\alpha_{i,1}^* + (1-\delta)\alpha_{i,2}^*)k(\mathbf{x}_i,\mathbf{x}) > \lambda_\delta \quad (16)$$

$$\sum_i \alpha_{i,2}^* k(\mathbf{x}_i,\mathbf{x}) > \lambda_2. \quad (17)$$

By $\frac{\epsilon}{\delta}\times$ (16) + $(1-\frac{\epsilon}{\delta})\times$ (17), we have $\sum_i(\epsilon\alpha_{i,1}^* + (1-\epsilon)\alpha_{i,2}^*)k(\mathbf{x}_i,\mathbf{x}) > \lambda_\epsilon$. Thus, $\widehat{G}_{\lambda_\delta} \subset \widehat{G}_{\lambda_\epsilon}$. ∎

The statement of this result focuses on the Gaussian kernel because this is the primary kernel for which the OC-SVM has been successfully applied.

C. Decomposition Algorithm

We also use a decomposition algorithm to solve (12). The general steps are the same as explained in Section III-C for the NCS-SVM. Fig. 4 shows the outline of the algorithm. In the algorithm, a feasible solution $\alpha_{i,m} = \frac{1}{N}$ for $\forall i,m$ is used as an initial solution.

Here we present how we can divide the large optimization problem into a collection of smaller problems. Suppose that the data point $\mathbf{x}_1$ is selected and its corresponding coefficients $\{\alpha_{1,m}\}_{m=1}^M$ will be updated. Writing the objective function only in terms of $\alpha_{1,m}$, we have

$$\begin{aligned}&\sum_m \left[\frac{1}{2\lambda_m}\sum_{i,j}\alpha_{i,m}\alpha_{j,m}K_{i,j} - \sum_i \alpha_{i,m} \right] \\ =&\sum_m \left[\frac{1}{2\lambda_m}\alpha_{1,m}^2K_{1,1} + \alpha_{1,m}\left(\frac{1}{\lambda_m}\sum_{j\ne1}\alpha_{j,m}K_{1,j} - 1\right)\right] + C \\ =&\sum_m \left[\frac{1}{2\lambda_m}\alpha_{1,m}^2K_{1,1} + \alpha_{1,m}\left(f_{1,m} - \frac{\alpha_{1,m}^{\text{old}}}{\lambda_m}K_{1,1} - 1\right)\right] + C \\ =&K_{1,1}\sum_m \left[\frac{1}{2\lambda_m}\alpha_{1,m}^2 - \frac{\alpha_{1,m}}{\lambda_m}\left(\alpha_{1,m}^{\text{old}} + \frac{\lambda_m(1-f_{1,m})}{K_{1,1}}\right)\right] + C\end{aligned}$$

where $\alpha_{1,m}^{\text{old}}$ and $f_{1,m} = \frac{1}{\lambda_m}\left(\sum_{j\ne1}\alpha_{j,m}K_{1,j} + \alpha_{1,m}^{\text{old}}K_{1,1}\right)$ denote the variable from the previous iteration step and the

Input: $\{\mathbf{x}_i\}_{i=1}^N$, $\{\lambda_m\}_{m=1}^M$

Initialize:

$$\alpha_{i,m} \leftarrow \frac{1}{N}, \quad \forall i, m$$

repeat

Choose a data point $\mathbf{x}_i$.

Compute:

$$f_{i,m} \leftarrow \frac{1}{\lambda_m} \sum_j \alpha_{j,m} K_{i,j}, \quad \forall m$$

$$\alpha_{i,m}^{\text{new}} \leftarrow \alpha_{i,m} + \frac{\lambda_m(1 - f_{i,m})}{K_{i,i}}, \quad \forall m$$

Update $\{\alpha_{i,m}\}_{m=1}^M$ with the solution of the subproblem:

$$\min_{\alpha_{i,1},\ldots,\alpha_{i,M}} \sum_m \left[\frac{1}{2\lambda_m} \alpha_{i,m}^2 - \frac{\alpha_{i,m}}{\lambda_m} \alpha_{i,m}^{\text{new}} \right]$$

$$\text{s.t.} \quad 0 \le \alpha_{i,m} \le \frac{1}{N}, \quad \forall m$$

$$\frac{\alpha_{i,1}}{\lambda_1} \le \frac{\alpha_{i,2}}{\lambda_2} \le \cdots \le \frac{\alpha_{i,M}}{\lambda_M}$$

until Accuracy conditions are satisfied

Output: $\widehat{G}_{\lambda_m} = \{\mathbf{x} : \sum_i \alpha_{i,m} k(\mathbf{x}_i, \mathbf{x}) > \lambda_m\}, \quad \forall m$

Fig. 4. Decomposition algorithm for a nested one-class SVM. Specific strategies for data point selection and termination, based on the KKT conditions, are given in the Appendix.

corresponding output, respectively. C is a constant that does not affect the solution.

Then we obtain the reduced optimization problem of M variables,

$$\min_{\alpha_{1,1},\ldots,\alpha_{1,M}} \sum_m \left[\frac{1}{2\lambda_m} \alpha_{1,m}^2 - \frac{\alpha_{1,m}}{\lambda_m} \alpha_{1,m}^{\text{new}} \right] \tag{18}$$

$$\text{s.t.} \quad 0 \le \alpha_{1,m} \le \frac{1}{N}, \quad \forall m \tag{19}$$

$$\frac{\alpha_{1,1}}{\lambda_1} \le \frac{\alpha_{1,2}}{\lambda_2} \le \cdots \le \frac{\alpha_{1,M}}{\lambda_M} \tag{20}$$

where $\alpha_{1,m}^{\text{new}} = \alpha_{1,m}^{\text{old}} + \frac{\lambda_m(1-f_{1,m})}{K_{1,1}}$. Notice that $\alpha_{1,m}^{\text{new}}$ becomes the solution if it is feasible. This reduced optimization problem can be solved through standard quadratic program solvers.

V. Computational Considerations

Here we provide guidelines for breakpoint selection and discuss the effects of interpolation.

A. Breakpoint Selection

The construction of an NCS-SVM begins with the selection of a finite number of cost asymmetries. Since the cost asymmetries take values within the range $[0, 1]$, the two breakpoints γ_1 and γ_M should be at the two extremes so that $\gamma_1 = 0$ and $\gamma_M = 1$. Then the rest of the breakpoints $\gamma_2, \cdots, \gamma_{M-1}$ can be set evenly spaced between γ_1 and γ_M.

On the other hand, the density levels for NOC-SVMs should be strictly positive. Without covering all positive reals, however, λ_1 and λ_M can be chosen to cover practically all the density levels of interest. The largest level λ_1 for the NOC-SVM is set as described in Appendix C where we show that for $\lambda > \lambda_1$, the CS-SVM and OC-SVM remain unchanged. A very small number greater than 0 is set for λ_M. Then the NOC-SVM is trained on evenly spaced breakpoints between λ_1 and λ_M.

In our experiments, we set the number of breakpoints to be $M = 5$ for NCS-SVMs and $M = 11$ for NOC-SVMs. These values were chosen because increasing the number of breakpoints M had diminishing AUC gains while causing training time increases in our experiments. Thus, the cost asymmetries for the NCS-SVM are $(0, 0.25, 0.5, 0.75, 1)$ and the density levels for NOC-SVM are 11 linearly spaced points from $\lambda_1 = \frac{1}{N} \max_i \sum_j K_{i,j}$ to $\lambda_{11} = 10^{-6}$.

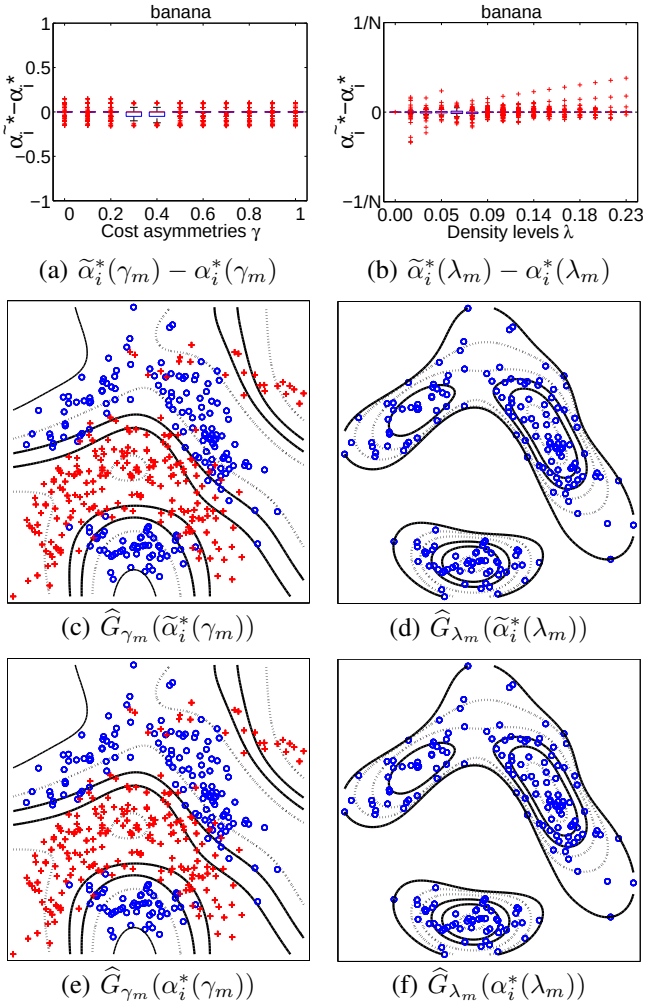
B. Effects of Interpolation

Nested SVMs are trained on a finite number of cost asymmetries or density levels and then the solution coefficients are linearly interpolated over a continuous range of parameters. Here we illustrate the effectiveness of the linear interpolation scheme of nested SVMs using the two dimensional `banana` data set.

Consider two sets of cost asymmetries, $\widetilde{\boldsymbol{\gamma}} = (0 : 0.25 : 1)$ and $\boldsymbol{\gamma} = (0 : 0.1 : 1)$, with different numbers of breakpoints for the NCS-SVM. Let $\widetilde{\alpha}_i^*(\gamma_m)$ denote the linearly interpolated solution at γ_m from the solution of the NCS-SVM with $\widetilde{\boldsymbol{\gamma}}$, and let $\alpha_i^*(\gamma_m)$ denote the solution from the NCS-SVM with $\boldsymbol{\gamma}$. Fig. 5 compares these two solution coefficients $\widetilde{\alpha}_i^*(\gamma_m)$ and $\alpha_i^*(\gamma_m)$. The box plots Fig. 5 (a) shows that values of $\widetilde{\alpha}_i^*(\gamma_m) - \alpha_i^*(\gamma_m)$ tend to be very small. Indeed, for most γ_m, the interquartile range on these box plots is not even visible. Regardless of these minor discrepancies, what is most important is that the resulting decision sets are almost indistinguishable as illustrated in Fig. 5 (c) and (e). Similar results can be observed in the NOC-SVM as well from Fig. 5 (b), (d) and (f). Here we consider two sets of density levels $\widetilde{\boldsymbol{\lambda}}$ with 11 breakpoints and $\boldsymbol{\lambda}$ with 16 breakpoints between $\lambda_1 = \frac{1}{N} \max_i \sum_j K_{i,j}$ and $\lambda_M = 10^{-6}$.

C. Computational complexity

According to Hastie et al. [11], the (non-nested) path following algorithm has $\mathcal{O}(N)$ breakpoints and complexity $\mathcal{O}(m^2N + N^2m)$, where m is the maximum number of points on the margin along the path. On the other hand, our nested SVMs have a controllable number of breakpoints M. To assess the complexity of the nested SVMs, we make a couple of assumptions based on experimental evidence. First, our experience has shown that the number of iterations of the decomposition algorithm is proportional to the number of data points N. Second, we assume that the subproblem, which has M variables, can be solved in $\mathcal{O}(M^2)$ operations. Furthermore, each iteration of the decomposition algorithm also involves a variable selection step. This involves checking all variables for KKT condition violations (as detailed in the Appendices), and thus entails $\mathcal{O}(MN)$ operations. Thus, the computation time of nested SVMs are $\mathcal{O}(M^2N + MN^2)$. In Section VI-E, we experimentally compare the run times of the path following algorithms to our methods.



(a) $\widetilde{\alpha}_i^*(\gamma_m) - \alpha_i^*(\gamma_m)$ (b) $\widetilde{\alpha}_i^*(\lambda_m) - \alpha_i^*(\lambda_m)$

(c) $\widehat{G}_{\gamma_m}(\widetilde{\alpha}_i^*(\gamma_m))$ (d) $\widehat{G}_{\lambda_m}(\widetilde{\alpha}_i^*(\lambda_m))$

(e) $\widehat{G}_{\gamma_m}(\alpha_i^*(\gamma_m))$ (f) $\widehat{G}_{\lambda_m}(\alpha_i^*(\lambda_m))$

Fig. 5. Simulation results depicting the impact of interpolation on the coefficients and final set estimates. See Section V-B for details.

Data set	dim	N_{train}	N_{test}
banana	2	400	4900
breast-cancer	9	200	77
diabetes	8	468	300
flare-solar	9	666	400
german	20	700	300
heart	13	170	100
ringnorm	20	400	7000
thyroid	5	140	75
titanic	3	150	2051
twonorm	20	400	7000
waveform	21	400	4600
image	18	1300	1010
splice	60	1000	2175

Fig. 6. Description of data sets. dim is the number of features, and N_{train} and N_{test} are the numbers of training and test examples.

VI. Experiments and Results

In order to compare the algorithms described above, we experimented on 13 benchmark data sets available online [1] [23]. Their brief summary is provided in Fig. 6. Each feature is standardized with zero mean and unit variance. The first eleven data sets are randomly permuted 100 times (the last two are permuted 20 times) and divided into training and test sets. In all of our experiments, we used the Gaussian kernel $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2}\right)$ and searched for the bandwidth σ over 20 logarithmically spaced points from $d_{avg}/15$ to $10\,d_{avg}$ where d_{avg} is the average distance between training data points. This control parameter is selected via 5-fold cross validation on the first 10 permutations, then the average of these values is used to train the remaining permutations.

Each algorithm generates a family of decision functions and set estimates. From these sets, we construct an ROC and compute its area under the curve (AUC). We use the AUC averaged across permutations to compare the performance of algorithms. As shown in Fig. 1, however, the set estimates from CS-SVMs or OC-SVMs are not properly nested, and cause ambiguity particularly in ranking. In Section VI-C, we measure this violation of the nesting by defining the *ranking disagreement* of two rank scoring functions. Then in Section VI-D, we combine this ranking disagreement and the AUC, and compare the algorithms over multiple data sets using the Wilcoxon signed ranks test as suggested in [24].

A. Two-class Problems

CS-SVMs and NCS-SVMs are compared in two-class problems. For NCS-SVMs, we set $M = 5$ and solved (7) at uniformly spaced cost asymmetries $\boldsymbol{\gamma} = (0, 0.25, 0.50, 0.75, 1)$.

In two-class problems, we also searched for the regularization parameter λ over 10 logarithmically spaced points from 0.1 to λ_{max} where λ_{max} is

$$\lambda_{max} = \max\left(\max_i \sum_{j\in I_+} y_i y_j K_{i,j},\ \max_i \sum_{j\in I_-} y_i y_j K_{i,j}\right).$$

Values of $\lambda > \lambda_{max}$ do not produce different solutions in the CS-SVM (see Appendix C).

We compared the described algorithms by constructing ROCs and computing their AUCs. The results are collected in Fig. 7. More statistical treatments of these results are covered in Section VI-D.

B. One-class Problems

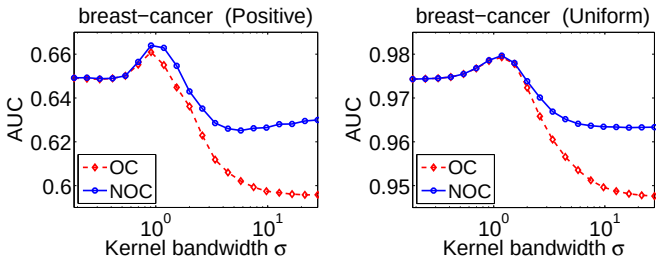
For the NOC-SVM, we selected 11 density levels spaced evenly from $\lambda_1 = \frac{1}{N}\max_i \sum_j K_{i,j}$ (see Appendix C) to $\lambda_{11} = 10^{-6}$. Among the two classes available in each data set, we chose the negative class for training. Because the bandwidth selection step requires computing AUCs, we simulated an artificial second class from a uniform distribution. For evaluation of the trained decision functions, both the positive examples in the test sets and a new uniform sample were used as the alternative class. Fig. 7 reports the results for both cases (denoted by Positive and Uniform, respectively).

Fig. 8 shows the AUC of the two algorithms over a range of σ. Throughout the experiments on one-class problems, we observed that the NOC-SVM is more robust to the kernel bandwidth selection than the OC-SVM. However, we did not observe similar results on two-class problems.

[1] http://ida.first.fhg.de/projects/bench/

Data Set	Two-class		One-class: Positive		One-class: Uniform	
	CS	NCS	OC	NOC	OC	NOC
banana	0.950 ± 0.009	0.963 ± 0.003	0.919 ± 0.009	0.930 ± 0.007	0.906 ± 0.003	0.911 ± 0.003
breast-cancer	0.733 ± 0.054	0.731 ± 0.056	0.647 ± 0.062	0.654 ± 0.061	0.976 ± 0.006	0.976 ± 0.006
diabetes	0.829 ± 0.016	0.825 ± 0.017	0.722 ± 0.023	0.732 ± 0.022	0.996 ± 0.001	0.996 ± 0.001
flare-solar	0.658 ± 0.040	0.580 ± 0.047	0.601 ± 0.042	0.601 ± 0.043	0.998 ± 0.000	0.998 ± 0.000
german	0.796 ± 0.024	0.788 ± 0.024	0.626 ± 0.031	0.626 ± 0.031	0.991 ± 0.003	0.991 ± 0.003
heart	0.908 ± 0.027	0.907 ± 0.027	0.776 ± 0.037	0.782 ± 0.036	0.986 ± 0.005	0.986 ± 0.005
ringnorm	0.982 ± 0.002	0.955 ± 0.011	0.997 ± 0.000	0.997 ± 0.000	1.000 ± 0.000	1.000 ± 0.000
thyroid	0.962 ± 0.037	0.954 ± 0.037	0.986 ± 0.009	0.987 ± 0.008	0.999 ± 0.000	0.999 ± 0.000
titanic	0.599 ± 0.069	0.597 ± 0.070	0.602 ± 0.068	0.588 ± 0.062	0.761 ± 0.051	0.765 ± 0.041
twonorm	0.997 ± 0.000	0.997 ± 0.000	0.910 ± 0.011	0.912 ± 0.009	1.000 ± 0.000	1.000 ± 0.000
waveform	0.969 ± 0.002	0.967 ± 0.003	0.752 ± 0.019	0.762 ± 0.017	1.000 ± 0.000	1.000 ± 0.000
image	0.991 ± 0.002	0.985 ± 0.004	0.872 ± 0.039	0.854 ± 0.039	1.000 ± 0.000	1.000 ± 0.000
splice	0.950 ± 0.003	0.951 ± 0.004	0.416 ± 0.009	0.415 ± 0.009	0.553 ± 0.012	0.554 ± 0.008

Fig. 7. AUC values for the CS-SVM (CS) and NCS-SVM (NCS) in two-class problems, and OC-SVM (OC) and NOC-SVM (NOC) in one-class problems. In one-class problems, 'Positive' indicates that the alternative hypotheses are from the positive class examples in the data sets, and 'Uniform' indicated that the alternative hypotheses are from a uniform distribution.



(a) Positive (b) Uniform

Fig. 8. The effect of kernel bandwidth σ on the performance (AUC). The AUC is evaluated when the alternative class is from the positive class in the data sets (a) and from a uniform distribution (b). The NOC-SVM is less sensitive to σ than the OC-SVM.

Data set	$d_2(s_+, s_-)$	$d_p(s_+, s_-)$	$d_u(s_+, s_-)$
banana	0.024	0.498	0.389
breast-cancer	0.013	0.252	0.093
diabetes	0.119	0.020	0.001
flare-solar	0.300	0.657	0.198
german	0.019	0.000	0.000
heart	0.005	0.000	0.000
ringnorm	0.244	0.000	0.000
thyroid	0.002	0.019	0.000
titanic	0.000	0.250	0.231
twonorm	0.006	0.000	0.000
waveform	0.078	0.002	0.001
image	0.307	0.276	0.047
splice	0.105	0.000	0.000

Fig. 9. The measure of disagreement of the two ranking functions from the CS-SVM and OC-SVM. The meaning of each subscript is explained in the text. s_+ and s_- are defined in (21) and (22).

C. Ranking disagreement

The decision sets from the OC-SVM and the CS-SVM are not properly nested, as illustrated in Fig. 1. Since larger λ means higher density level, the density level set estimate of the OC-SVM is expected to be contained within the density level set estimate at smaller λ. Likewise, larger γ in the CS-SVM penalizes misclassification of positive examples more; thus, its corresponding positive decision set should contain the decision set at smaller γ, and the two decision boundaries should not cross. This undesired nature of the algorithms leads to non-unique ranking score functions.

In the case of the CS-SVM, we can consider the following two ranking functions:

$$s_+(\mathbf{x}) = 1 - \min_{\{\gamma: f_\gamma(\mathbf{x}) \geq 0\}} \gamma, \quad s_-(\mathbf{x}) = 1 - \max_{\{\gamma: f_\gamma(\mathbf{x}) \leq 0\}} \gamma. \tag{21}$$

For the OC-SVM, we consider the next pair of ranking functions,

$$s_+(\mathbf{x}) = \max_{\{\lambda: \mathbf{x} \in \widehat{G}_\lambda\}} \lambda, \qquad s_-(\mathbf{x}) = \min_{\{\lambda: \mathbf{x} \in \widehat{G}_\lambda\}} \lambda. \tag{22}$$

In words, s_+ ranks according to the first set containing a point $\mathbf{x}$ and s_- ranks according to the last set containing the point. In either case, it is easy to see $s_+(\mathbf{x}) \geq s_-(\mathbf{x})$.

In order to quantify the disagreement of the two ranking functions, we define the following measure of *ranking disagreement*:

$$d(s_+, s_-) = \frac{1}{N} \sum_i \max_{j \neq i} \mathbf{1}_{\{(s_+(\mathbf{x}_i) - s_+(\mathbf{x}_j))(s_-(\mathbf{x}_i) - s_-(\mathbf{x}_j)) < 0\}},$$

which is the proportion of data points ambiguously ranked, i.e., ranked differently with respect to at least one other point. Then $d(s_+, s_-) = 0$ if and only if s_+ and s_- induce the same ranking.

With these ranking functions, Fig. 9 reports the ranking disagreements from the CS-SVM and OC-SVM. In the table, d_2 refers to the ranking disagreement of the CS-SVM, and d_p and d_u respectively refer to the ranking disagreement of the OC-SVM when the second class is from the positive samples and from an artificial uniform distribution. As can be seen in the table, for some data sets the violation of the nesting causes severe differences between the above ranking functions.

D. Statistical comparison

We employ the statistical methodology of Demšar [24] to compare the algorithms across all data sets. Using the Wilcoxon signed ranks test, we compare the CS-SVM and the NCS-SVM for two-class problems, and the OC-SVM and the NOC-SVM for one-class problems.

The Wilcoxon signed ranks test is a non-parametric method testing the significance of differences between paired observations, and can be used to compare the performances between

CS	NCS	T
78	13	13

Fig. 10. Comparison of the AUCs of the two-class problem algorithms: CS-SVM (CS) and NCS-SVM (NCS) using the Wilcoxon signed ranks test (see text for detail.) The test statistic T is greater than the critical difference 9, hence no significant difference is detected in the test.

	OC	NOC	T
Positive	35	56	35
Uniform	21.5	69.5	21.5

Fig. 11. Comparison of the OC-SVM (OC) and NOC-SVM (NOC). In the one-class problems, both cases of alternative hypothesis are considered. Here no significant difference is detected.

two algorithms over multiple data sets. The difference between the AUCs from the two algorithms are ranked ignoring the signs, and then the ranks of positive and negative differences are added. Fig. 10 and Fig. 11 respectively report the comparison results of the algorithms for two-class problems and one-class problems. Here the numbers under NCS or NOC denote the sums of ranks of the data sets on which the nested SVMs performed better than the original SVMs; the values under CS or OC are for the opposite. T is the smaller of the two sums. For a confidence level of $\alpha = 0.01$ and 13 data sets, the difference between algorithms is significant if T is less than or equal to 9 [25]. Therefore, any significant performance difference between the CS-SVM and the NCS-SVM was not detected in the test. Likewise, no difference between the OC-SVM and the NOC-SVM was detected.

However, the AUC alone does not highlight the ranking disagreement of the algorithms. Therefore, we merge the AUC and the disorder measurement, and consider $AUC - d(s_+, s_-)$ for algorithm comparison. Fig. 12 shows the results of the Wilcoxon signed-ranks test using this combined performance measure. From the results, we can observe clearly the performance differences between algorithms. Since the test statistic T is smaller than the critical difference 9, the NCS-SVM outperforms the CS-SVM. Likewise, the performance difference between the OC-SVM and the NOC-SVM is also detected by the Wilcoxon test for both cases of the second class. Therefore, we can conclude that the nested algorithms perform better than their unnested counterparts.

E. Run time comparison

Fig. 13 shows the training times for each algorithm. The results for the CS-SVM and OC-SVM are based on our Matlab implementation of solution path algorithms [8], [12] available at http://www.eecs.umich.edu/~cscott/code/svmpath.zip. We emphasize here that our decomposition algorithm relies on Matlab's `quadprog` function as the basic subproblem solver, and that this function is in no way optimized for our particular subproblem. A discussion of computational complexity was given in V-C.

CS	NCS	T
4	87	4

	OC	NOC	T
Positive	5	86	5
Uniform	2.5	88.5	2.5

Fig. 12. Comparison of the algorithms based on the AUC along with the *ranking disagreement*. **Left**: CS-SVM and NCS-SVM. **Right**: OC-SVM and NOC-SVM. T is less than the critical values 9, hence the nested SVMs outperforms the original SVMs.

Data set	CS	NCS	OC	NOC
banana	1.01	24.55	0.29	13.03
breast-cancer	0.43	2.42	0.13	9.64
diabetes	2.92	9.80	0.56	75.46
flare-solar	0.17	4.05	0.02	0.85
german	13.25	0.68	4.48	57.69
heart	0.31	7.76	0.07	5.71
ringnorm	3.16	3.43	0.01	2.07
thyroid	0.22	2.74	0.08	6.50
titanic	0.01	0.66	< 0.01	5.69
twonorm	1.89	8.21	0.31	15.29
waveform	1.87	10.42	0.56	26.60
image	40.08	298.98	1.30	64.77
splice	68.43	149.68	0.55	6.06

Fig. 13. Average training times (sec) for the CS-SVM, NCS-SVM, OC-SVM, and NOC-SVM on benchmark data sets. This result is based on our implementation of solution path algorithms for the CS-SVM and OC-SVM.

VII. Primal of Nested SVMs

Although not essential for our approach, we can find a primal optimization problem of the NCS-SVM if we think of (7) as a dual problem:

$$\begin{aligned}
\min_{\mathbf{w},\boldsymbol{\xi}} & \sum_{m=1}^{M} \left[\frac{\lambda}{2}\|\mathbf{w}_m\|^2 + \gamma_m \sum_{I_+} \xi_{i,m} + (1-\gamma_m) \sum_{I_-} \xi_{i,m} \right] \\
\text{s.t.} \quad & \sum_{k=m}^{M} \langle \mathbf{w}_k, \Phi(\mathbf{x}_i) \rangle \geq \sum_{k=m}^{M} (1-\xi_{i,k}), \quad i \in I_+, \forall m \\
& \sum_{k=1}^{m} \langle \mathbf{w}_k, \Phi(\mathbf{x}_i) \rangle \leq -\sum_{k=1}^{m} (1-\xi_{i,k}), \quad i \in I_-, \forall m \\
& \xi_{i,m} \geq 0, \quad \forall i, m.
\end{aligned}$$

The derivation of (7) from this primal can be found in [26]. Note that the above primal of the NCS-SVM reduces to the primal of the CS-SVM (1) when $M = 1$.

Likewise, the primal corresponding to the NOC-SVM is

$$\begin{aligned}
\min_{\mathbf{w},\boldsymbol{\xi}} & \sum_{m=1}^{M} \left[\frac{\lambda_m}{2}\|\mathbf{w}_m\|^2 + \frac{1}{N} \sum_{i} \xi_{i,m} \right] \\
\text{s.t.} \quad & \sum_{k=m}^{M} \lambda_k \langle \mathbf{w}_k, \Phi(\mathbf{x}_i) \rangle \geq \sum_{k=m}^{M} \lambda_k (1-\xi_{i,m}), \quad \forall i, m \\
& \xi_{i,m} \geq 0, \quad \forall i, m,
\end{aligned} \tag{23}$$

which also boils down to the primal of the OC-SVM (4) when $M = 1$.

With these formulations, we can see the geometric meaning of $\mathbf{w}$ and $\boldsymbol{\xi}$. For simplicity, consider (23) when $M = 2$:

$$\begin{aligned}
\min_{\mathbf{w},\boldsymbol{\xi}} & \frac{\lambda_2}{2}\|\mathbf{w}_2\|^2 + \frac{1}{N}\sum_i \xi_{i,2} + \frac{\lambda_1}{2}\|\mathbf{w}_1\|^2 + \frac{1}{N}\sum_i \xi_{i,1} \\
\text{s.t.} \quad & \langle \lambda_2 \mathbf{w}_2, \Phi(\mathbf{x}_i) \rangle \geq \lambda_2 (1-\xi_{i,2}), \quad \forall i \\
& \langle \lambda_2 \mathbf{w}_2 + \lambda_1 \mathbf{w}_1, \Phi(\mathbf{x}_i) \rangle \geq \lambda_2 (1-\xi_{i,2}) + \lambda_1 (1-\xi_{i,1}), \quad \forall i \\
& \xi_{i,m} \geq 0, \quad \forall i, m.
\end{aligned}$$

Here $\xi_{i,1} > 0$ when $\mathbf{x}_i$ lies between the hyperplane $P_{\frac{\lambda_2 \mathbf{w}_2 + \lambda_1 \mathbf{w}_1}{\lambda_2 + \lambda_1}}$ and the origin, and $\xi_{i,2} > 0$ when the point lies

between $P_{\mathbf{w}_2}$ and the origin where we used $P_{\mathbf{w}}$ to denote $\{\Phi(\mathbf{x}) : \langle \mathbf{w}, \Phi(\mathbf{x})\rangle = 1\}$, a hyperplane in $\mathcal{H}$. Note that from the nesting structure, the hyperplane $P_{\frac{\lambda_2\mathbf{w}_2+\lambda_1\mathbf{w}_1}{\lambda_2+\lambda_1}}$ is located between $P_{\mathbf{w}_1}$ and $P_{\mathbf{w}_2}$. Then we can show that $\frac{\lambda_1\xi_{i,1}+\lambda_2\xi_{i,2}}{\|\lambda_1\mathbf{w}_1+\lambda_2\mathbf{w}_2\|}$ is the distance between the point $\mathbf{x}_i$ and the hyperplane $P_{\frac{\lambda_2\mathbf{w}_2+\lambda_1\mathbf{w}_1}{\lambda_2+\lambda_1}}$.

VIII. Conclusion

In this paper, we introduced a novel framework for building a family of nested support vector machines for the tasks of cost-sensitive classification and density level set estimation. Our approach involves forming new quadratic programs inspired by the cost-sensitive and one-class SVMs, with additional constraints that enforce nesting structure. Our construction generates a finite number of nested set estimates at a preselected set of parameter values, and linearly interpolates these sets to a continuous nested family. We also developed efficient algorithms to solve the proposed quadratic problems. Thus, the NCS-SVM yields a family of nested classifiers indexed by cost asymmetry γ, and the NOC-SVM yields a family of nested density level set estimates indexed by density level λ. Unlike the original SVMs, which are not nested, our methods can be readily applied to problems requiring multiple set estimation including clustering, ranking, and anomaly detection.

In experimental evaluations, we found that non-nested SVMs can yield highly ambiguous rankings for many datasets, and that nested SVMs offer considerable improvements in this regard. Nested SVMs also exhibit greater stability with respect to model selection criteria such as cross-validation. In terms of area under the ROC (AUC), we found that enforcement of nesting appears to have a bigger impact on one-class problems. However, neither cost-sensitive nor one-class classification problems displayed significantly different AUC values between nested and non-nested methods.

The statistical consistency of our nested SVMs is an interesting open question. Such a result would likely depend on the consistency of the original CS-SVM or OC-SVM at fixed values of γ or λ, respectively. We are unaware of consistency results for the CS-SVM at fixed γ [27]. However, consistency of the OC-SVM for fixed λ has been established [20]. Thus, suppose $\widehat{G}_{\lambda_1}, \ldots, \widehat{G}_{\lambda_M}$ are (non-nested) OC-SVMs at a grid of points. Since these estimators are each consistent, and the true levels sets they approximate are nested, it seems plausible that for a sufficiently large sample size, these OC-SVMs are also nested. In this case, they would be feasible for the NOC-SVM, which would suggest that the NOC-SVM estimates the true level sets at least as well, asymptotically, at these estimates. Taking the grid of levels $\{\lambda_i\}$ to be increasingly dense, the error of the interpolation scheme should also vanish. We leave it as future work to determine whether this intuition can be formalized.

Appendix A
Data Point Selection and Termination Condition of NCS-SVM

On each round, the algorithm in Fig. 3 selects an example $\mathbf{x}_i$, updates its corresponding variables $\{\alpha_{i,m}\}_{m=1}^M$, and checks the termination condition. In this appendix, we employ the KKT conditions to derive an efficient variable selection strategy and a termination condition of NCS-SVM.

We use the KKT conditions to find the necessary conditions of the optimal solution of (7). Before we proceed, we define $\alpha_{i,0} = 0$ for $i \in I_+$ and $\alpha_{i,M+1} = 0$ for $i \in I_-$ for notational convenience. Then the Lagrangian of the quadratic program is

$$\begin{aligned}\mathcal{L}(\boldsymbol{\alpha}, \mathbf{u}, \mathbf{v}) = &\sum_m \left[\frac{1}{2\lambda}\sum_{i,j}\alpha_{i,m}\alpha_{j,m}y_iy_jK_{i,j} - \sum_i \alpha_{i,m}\right] \\ &+ \sum_m\sum_i u_{i,m}(\alpha_{i,m} - \mathbf{1}_{\{y_i<0\}} - y_i\gamma_m) \\ &+ \sum_m\sum_{i\in I_+} v_{i,m}(\alpha_{i,m-1} - \alpha_{i,m}) \\ &- \sum_m\sum_{i\in I_-} v_{i,m}(\alpha_{i,m} - \alpha_{i,m+1})\end{aligned}$$

where $u_{i,m} \geq 0$ and $v_{i,m} \geq 0$ for $\forall i, m$. At the global minimum, the derivative of the Lagrangian with respect to $\alpha_{i,m}$ vanishes

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\alpha_{i,m}} &= y_if_{i,m} - 1 + u_{i,m}\begin{cases}-v_{i,m} + v_{i,m+1}, & i \in I_+ \\ +v_{i,m-1} - v_{i,m}, & i \in I_-\end{cases} \\ &= 0 \end{aligned} \tag{24}$$

where, recall, $f_{i,m} = \frac{1}{\lambda}\sum_j \alpha_{j,m}y_jK_{i,j}$ and we introduced auxiliary variables $v_{i,M+1} = 0$ for $i \in I_+$ and $v_{i,0} = 0$ for $i \in I_-$. Then we obtain the following set of constraints from the KKT conditions

$$y_if_{i,m} - 1 + u_{i,m} = \begin{cases}v_{i,m} - v_{i,m+1}, & i \in I_+ \\ -v_{i,m-1} + v_{i,m}, & i \in I_-\end{cases} \tag{25}$$

$$0 \leq \alpha_{i,m} \leq \mathbf{1}_{\{y_i<0\}} + y_i\gamma_m, \qquad \forall i, \forall m \tag{26}$$

$$y_i\alpha_{i,1} \leq y_i\alpha_{i,2} \leq \cdots \leq y_i\alpha_{i,M}, \qquad \forall i \tag{27}$$

$$u_{i,m}\left(\alpha_{i,m} - \mathbf{1}_{\{y_i<0\}} - y_i\gamma_m\right) = 0, \qquad \forall i, \forall m \tag{28}$$

$$v_{i,m}(\alpha_{i,m-1} - \alpha_{i,m}) = 0, \qquad i \in I_+, \forall m \tag{29}$$

$$v_{i,m}(\alpha_{i,m} - \alpha_{i,m+1}) = 0, \qquad i \in I_-, \forall m \tag{30}$$

$$u_{i,m} \geq 0, \qquad v_{i,m} \geq 0, \qquad \forall i, m. \tag{31}$$

Since (7) is a convex program, the KKT conditions are also sufficient [22]. That is, $\alpha_{i,m}$, $u_{i,m}$, and $v_{i,m}$ satisfying (25)-(31) is indeed optimal. Therefore, at the end of each iteration, we assess a current solution with these conditions and decide whether to stop or to continue. We evaluate the amount of error for $\mathbf{x}_i$ by defining

$$e_i = \sum_m \left|\frac{\partial\mathcal{L}}{\partial\alpha_{i,m}}\right|, \quad \forall i.$$

An optimal solution makes these quantities zero. In practice, when their sum $\sum_i e_i$ decreases below a predetermined tolerance, the algorithm stops and returns the current solution. If not, the algorithm chooses the example with the largest e_i and continues the loop.

Computing e_i involves unknown variables $u_{i,m}$ and $v_{i,m}$ (see (24)), whereas $f_{i,m}$ can be easily computed from the known variables $\alpha_{i,m}$. Fig. 14 and Fig. 15 are for determining

	$\alpha_{i,m-1} < \alpha_{i,m}$	$\alpha_{i,m-1} = \alpha_{i,m}$
$\alpha_{i,m} < \min(\gamma_m, \alpha_{i,m+1})$	$u_{i,m} = 0$ $v_{i,m} = 0$	$u_{i,m} = 0$ $v_{i,m} = \max(f_{i,m} - 1, 0)$
$\alpha_{i,m} = \gamma_m < \alpha_{i,m+1}$	$u_{i,m} = \max(1 - f_{i,m}, 0)$ $v_{i,m} = 0$	- -
$\alpha_{i,m} = \alpha_{i,m+1} < \gamma_m$	$u_{i,m} = 0$ $v_{i,m} = 0$	$u_{i,m} = 0$ $v_{i,m} = \max(f_{i,m} - 1 + v_{i,m+1}, 0)$
$\alpha_{i,m} = \alpha_{i,m+1} = \gamma_m$	$u_{i,m} = \max(1 - f_{i,m} - v_{i,m+1}, 0)$ $v_{i,m} = 0$	- -

	$\alpha_{i,M-1} < \alpha_{i,M}$	$\alpha_{i,M-1} = \alpha_{i,M}$
$\alpha_{i,M} < \gamma_M$	$u_{i,M} = 0$ $v_{i,M} = 0$	$u_{i,M} = 0$ $v_{i,M} = \max(f_{i,M} - 1, 0)$
$\alpha_{i,M} = \gamma_M$	$u_{i,M} = \max(1 - f_{i,M}, 0)$ $v_{i,M} = 0$	- -

Fig. 14. The optimality conditions of NCS-SVM when $i \in I_+$. (Upper: $m = 1, 2, \ldots, M-1$, Lower: $m = M$.) Assuming $\alpha_{i,m}$ are optimal, $u_{i,m}$ and $v_{i,m}$ are solved as above from the KKT conditions. Empty entries indicate cases that cannot occur.

	$\alpha_{i,m+1} < \alpha_{i,m}$	$\alpha_{i,m+1} = \alpha_{i,m}$
$\alpha_{i,m} < \min(1 - \gamma_m, \alpha_{i,m-1})$	$u_{i,m} = 0$ $v_{i,m} = 0$	$u_{i,m} = 0$ $v_{i,m} = \max(-f_{i,m} - 1, 0)$
$\alpha_{i,m} = 1 - \gamma_m < \alpha_{i,m-1}$	$u_{i,m} = \max(1 + f_{i,m}, 0)$ $v_{i,m} = 0$	- -
$\alpha_{i,m} = \alpha_{i,m-1} < 1 - \gamma_m$	$u_{i,m} = 0$ $v_{i,m} = 0$	$u_{i,m} = 0$ $v_{i,m} = \max(-f_{i,m} - 1 + v_{i,m-1}, 0)$
$\alpha_{i,m} = \alpha_{i,m-1} = 1 - \gamma_m$	$u_{i,m} = \max(1 + f_{i,m} - v_{i,m-1}, 0)$ $v_{i,m} = 0$	- -

	$\alpha_{i,2} < \alpha_{i,1}$	$\alpha_{i,2} = \alpha_{i,1}$
$\alpha_{i,1} < 1 - \gamma_1$	$u_{i,1} = 0$ $v_{i,1} = 0$	$u_{i,1} = 0$ $v_{i,1} = \max(-f_{i,1} - 1, 0)$
$\alpha_{i,1} = 1 - \gamma_1$	$u_{i,1} = \max(1 + f_{i,1}, 0)$ $v_{i,1} = 0$	- -

Fig. 15. The optimality conditions of NCS-SVM when $i \in I_-$. (Upper: $m = 2, \ldots, M$, Lower: $m = 1$.)

these $u_{i,m}$ and $v_{i,m}$. These tables are obtained by firstly assuming the current solution $\alpha_{i,m}$ is optimal and secondly solving $u_{i,m}$ and $v_{i,m}$ such that they satisfy the KKT conditions. Thus, depending on the value $\alpha_{i,m}$ between its upper and lower bounds, $u_{i,m}$ and $v_{i,m}$ can be simply set as directed in the tables. For example, if $i \in I_+$, then we find $u_{i,m}$ and $v_{i,m}$ by referring Fig. 14 iteratively from $m = M$ down to $m = 1$. If $i \in I_-$, we use Fig. 15 and iterate from $m = 1$ up to $m = M$. Then the obtained e_i takes a non-zero value only when the assumption is false and the current solution is sub-optimal.

Appendix B
Data Point Selection and Termination Condition of NOC-SVM

As in NCS-SVM, we investigate the optimality condition of NOC-SVM (12) and find a data point selection method and a termination condition.

With a slight modification, we rewrite (12),

$$\min_{\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_M} \sum_{m=1}^{M} \left[\frac{1}{2\lambda_m} \sum_{i,j} \alpha_{i,m}\alpha_{j,m}K_{i,j} - \sum_i \alpha_{i,m} \right] \quad (32)$$
$$\text{s.t.} \quad \alpha_{i,m} \le \frac{1}{N}, \quad \forall i, m$$
$$0 \le \frac{\alpha_{i,1}}{\lambda_1} \le \frac{\alpha_{i,2}}{\lambda_2} \le \cdots \le \frac{\alpha_{i,M}}{\lambda_M}, \quad \forall i.$$

We then use the KKT conditions to find the necessary conditions of the optimal solution of (32). The Lagrangian is

$$\begin{aligned}\mathcal{L}(\boldsymbol{\alpha}, \mathbf{u}, \mathbf{v}) = &\sum_{m=1}^{M} \left[\frac{1}{2\lambda_m} \sum_{i,j} \alpha_{i,m}\alpha_{j,m}K_{i,j} - \sum_i \alpha_{i,m} \right] \\ &+ \sum_{m=1}^{M} \sum_i u_{i,m}(\alpha_{i,m} - \frac{1}{N}) - \sum_i v_{i,1}\frac{\alpha_{i,1}}{\lambda_1} \\ &+ \sum_i \sum_{m=2}^{M} v_{i,m}\left(\frac{\alpha_{i,m-1}}{\lambda_{m-1}} - \frac{\alpha_{i,m}}{\lambda_m} \right)\end{aligned}$$

where $u_{i,m} \ge 0$ and $v_{i,m} \ge 0$ for $\forall i, m$. At the global minimum, the derivative of the Lagrangian with respect to $\alpha_{i,m}$ vanishes

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \alpha_{i,m}} &= f_{i,m} - 1 + u_{i,m} \begin{cases} -\frac{v_{i,m}}{\lambda_m} + \frac{v_{i,m+1}}{\lambda_m}, & m \ne M \\ -\frac{v_{i,M}}{\lambda_M}, & m = M \end{cases} \\ &= 0. \end{aligned} \quad (33)$$

where, recall, $f_{i,m} = \frac{1}{\lambda_m}\sum_j \alpha_{j,m}K_{i,j}$. Then, from the KKT

conditions, we obtain the following set of constraints for $\mathbf{x}_i$:

$$f_{i,m} - 1 + u_{i,m} = \begin{cases} \frac{v_{i,m}}{\lambda_m} - \frac{v_{i,m+1}}{\lambda_m}, & m \neq M \\ \frac{v_{i,M}}{\lambda_M}, & m = M \end{cases} \tag{34}$$

$$\alpha_{i,m} \leq \frac{1}{N}, \qquad \forall m \tag{35}$$

$$0 \leq \frac{\alpha_{i,1}}{\lambda_1} \leq \frac{\alpha_{i,2}}{\lambda_2} \leq \cdots \leq \frac{\alpha_{i,M}}{\lambda_M} \tag{36}$$

$$u_{i,m}(\alpha_{i,m} - \frac{1}{N}) = 0, \qquad \forall m \tag{37}$$

$$v_{i,m}(\frac{\alpha_{i,m-1}}{\lambda_{m-1}} - \frac{\alpha_{i,m}}{\lambda_m}) = 0, \qquad \forall m \tag{38}$$

$$u_{i,m} \geq 0, \qquad v_{i,m} \geq 0, \qquad \forall m. \tag{39}$$

Since (32) is a convex program, the KKT conditions are sufficient [22]. That is, $\alpha_{i,m}$, $u_{i,m}$, and $v_{i,m}$ satisfying (34)-(39) is indeed optimal. Therefore, at the end of each iteration, we assess a current solution with these conditions and decide whether to stop or to continue. We evaluate the amount of error for $\mathbf{x}_i$ by defining

$$e_i = \sum_m \left| \frac{\partial \mathcal{L}}{\partial \alpha_{i,m}} \right|, \quad \forall i.$$

An optimal solution makes these quantities zero. In practice, when their sum $\sum_i e_i$ decreases below a predetermined tolerance, the algorithm stops and returns the current solution. If not, the algorithm chooses the example with the largest e_i and continues the loop.

Computing e_i involves unknown variables $u_{i,m}$ and $v_{i,m}$ (see (33)), whereas $f_{i,m}$ can be easily computed from the known variables $\alpha_{i,m}$. Fig. 16 are for determining these $u_{i,m}$ and $v_{i,m}$. These tables are obtained by firstly assuming the current solution $\alpha_{i,m}$ is optimal and secondly solving $u_{i,m}$ and $v_{i,m}$ such that they satisfy the KKT conditions. Thus, depending on the value $\alpha_{i,m}$ between its upper and lower bounds, $u_{i,m}$ and $v_{i,m}$ can be simply set by referring Fig. 16 iteratively from $m = M$ down to $m = 1$. Then the obtained e_i takes a non-zero value only when the assumption is false and the current solution is not optimal.

APPENDIX C
MAXIMUM VALUE OF λ OF CS-SVM AND OC-SVM

In this appendix, we find the values of the regularization parameter λ over which OC-SVM or CS-SVM generate the same solutions.

First, we consider OC-SVM. The decision function of OC-SVM is $f_\lambda(\mathbf{x}) = \frac{1}{\lambda}\sum_j \alpha_j k(\mathbf{x}_j, \mathbf{x})$ and $f_\lambda(\mathbf{x}) = 1$ forms the margin. For sufficiently large λ, every data point $\mathbf{x}_i$ falls inside the margin ($f_\lambda(\mathbf{x}_i) \leq 1$). Since the KKT optimality conditions of (4) imply $\alpha_i = \frac{1}{N}$ for the data points such that $f_\lambda(\mathbf{x}_i) < 1$, we obtain $\lambda \geq \frac{1}{N}\sum_j K_{i,j}$ for $\forall i$. Therefore, if the maximum row sum of the kernel matrix is denoted as $\lambda_{OC} = \max_i \frac{1}{N}\sum_j K_{i,j}$, then for any $\lambda \geq \lambda_{OC}$, the optimal solution of OC-SVM becomes $\alpha_i = \frac{1}{N}$ for $\forall i$.

Next, we consider the regularization parameter λ of in the formulation (1) of CS-SVM. The decision function of CS-SVM is $f_\gamma(\mathbf{x}) = \frac{1}{\lambda}\sum_j \alpha_j y_j k(\mathbf{x}_j, \mathbf{x})$, and the margin is $y f_\gamma(\mathbf{x}) = 1$. Thus, if λ is sufficiently large, all the data points are inside the margin and satisfy $y_i f_\gamma(\mathbf{x}_i) \leq 1$. Then $\lambda \geq \sum_{j\in I_+} \gamma y_i y_j K_{i,j} + \sum_{j\in I_-}(1-\gamma) y_i y_j K_{i,j}$ for $\forall i$ because $\alpha_i = \mathbf{1}_{\{y_i<0\}} + y_i\gamma$ for all the data points such that $y_i f_\gamma(\mathbf{x}_i) < 1$ from the KKT conditions. For a given γ, let

$$\lambda_{CS}(\gamma) = \max_i \left[\gamma \sum_{j\in I_+} y_i y_j K_{i,j} + (1-\gamma) \sum_{j\in I_-} y_i y_j K_{i,j} \right].$$

Then for $\lambda > \lambda_{CS}(\gamma)$, the solution of CS-SVM becomes $\alpha_i = \mathbf{1}_{\{y_i<0\}} + y_i\gamma$ for $\forall i$. Therefore, since $\lambda_{CS}(\gamma) \leq (1-\gamma)\lambda_{CS}(0) + \gamma\lambda_{CS}(1)$ for all $\gamma \in [0,1]$, values of $\lambda > \max(\lambda_{CS}(0), \lambda_{CS}(1))$ generate the same solutions in CS-SVM.

REFERENCES

[1] J. A. Hartigan, "Consistency of single linkage for high-density clusters," *J. of the American Stat. Association*, vol. 76, pp. 388–394, 1981.

[2] R. Liu, J. Parelius, and K. Singh, "Multivariate analysis by data depth: descriptive statistics, graphics and inference," *Annals of Statistics*, vol. 27, pp. 783–858, 1999.

[3] C. Scott and R. Nowak, "Learning minimum volume sets," *Journal of Machine Learning Research*, vol. 7, pp. 665–704, 2006.

[4] C. Scott and E. D. Kolaczyk, "Annotated minimum volume sets for nonparametric anomaly discovery," in *IEEE Workshop on Statistical Signal Processing*, 2007, pp. 234–238.

[5] R. Herbrich, T. Graepel, and K. Obermayer, "Large margin rank boundaries for ordinal regression," *Advances in Large Margin Classifiers*, pp. 115–132, 2000.

[6] C. Scott and R. Nowak, "A Neyman-Pearson approach to statistical learning," *IEEE Trans. Inf. Theory*, vol. 51, pp. 3806–3819, 2005.

[7] C. Scott and G. Blanchard, "Novelty detection: Unlabeled data definitely help," *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, vol. 5, pp. 464–471, 2009.

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

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

[10] S. J. Clémençon and N. Vayatis, "Overlaying classifiers: a practical approach for optimal ranking," *Advances in Neural Information Processing Systems 21*, vol. 21, pp. 313–320, 2009.

[11] T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu, "The entire regularization path for the support vector machine," *Journal of Machine Learning Research*, vol. 5, pp. 1391–1415, 2004.

[12] G. Lee and C. Scott, "The one class support vector machine solution path," in *IEEE Intl. Conf. on Acoustics, Speech and Signal Proc. (ICASSP)*, vol. 2, 2007, pp. II–521–II–524.

[13] ——, "Nested support vector machines," in *IEEE Intl. Conf. on Acoustics, Speech and Signal Proc. (ICASSP)*, 2008, pp. 1985–1988.

[14] S. Agarwal, T. Graepel, R. Herbrich, S. Har-Peled, and D. Roth, "Generalization bounds for the area under the roc curve," *Journal of Machine Learning Research*, vol. 6, pp. 393–425, 2005.

[15] W. Stuetzle, "Estimating the cluster tree of a density by analyzing the minimal spanning tree of a sample," *Journal of Classification*, vol. 20, no. 5, pp. 25–47, 2003.

[16] V. Kecman, *Learning and Soft Computing, Support Vector Machines, Neural Networks, and Fuzzy Logic Models*. Cambridge, MA: MIT Press, 2001.

[17] E. Osuna, R. Freund, and F. Girosi, "Support vector machines: Training and applications," MIT Artificial Intelligence Laboratory, Tech. Rep. AIM-1602, Mar 1997.

[18] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, pp. 1443–1472, 2001.

[19] D. Tax and R. Duin, "Support vector domain description," *Pattern Recognition Letters*, vol. 20, pp. 1191–1199, 1999.

[20] R. Vert and J. Vert, "Consistency and convergence rates of one-class SVMs and related algorithms," *Journal of Machine Learning Research*, vol. 7, pp. 817–854, 2006.

[21] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *Journal of Machine Learning Research*, vol. 2, pp. 265–292, 2001.

	$\frac{\lambda_m}{\lambda_{m-1}}\alpha_{i,m-1} < \alpha_{i,m}$	$\frac{\lambda_m}{\lambda_{m-1}}\alpha_{i,m-1} = \alpha_{i,m}$
$\alpha_{i,m} < \min(\frac{1}{N}, \frac{\lambda_m}{\lambda_{m+1}}\alpha_{i,m+1})$	$u_{i,m} = 0$ $v_{i,m} = 0$	$u_{i,m} = 0$ $v_{i,m} = \max(\lambda_m(f_{i,m} - 1), 0)$
$\alpha_{i,m} = \frac{1}{N} < \frac{\lambda_m}{\lambda_{m+1}}\alpha_{i,m+1}$	$u_{i,m} = \max(1 - f_{i,m}, 0)$ $v_{i,m} = 0$	- -
$\alpha_{i,m} = \frac{\lambda_m}{\lambda_{m+1}}\alpha_{i,m+1} < \frac{1}{N}$	$u_{i,m} = 0$ $v_{i,m} = 0$	$u_{i,m} = 0$ $v_{i,m} = \max(\lambda_m(f_{i,m} - 1 + \frac{v_{i,m+1}}{\lambda_m}), 0)$
$\alpha_{i,m} = \frac{\lambda_m}{\lambda_{m+1}}\alpha_{i,m+1} = \frac{1}{N}$	$u_{i,m} = \max(1 - f_{i,m} - \frac{v_{i,m+1}}{\lambda_m}, 0)$ $v_{i,m} = 0$	- -

	$\frac{\lambda_M}{\lambda_{M-1}}\alpha_{i,M-1} < \alpha_{i,M}$	$\frac{\lambda_M}{\lambda_{M-1}}\alpha_{i,M-1} = \alpha_{i,M}$
$\alpha_{i,M} < \frac{1}{N}$	$u_{i,M} = 0$ $v_{i,M} = 0$	$u_{i,M} = 0$ $v_{i,M} = \max(\lambda_M(f_{i,M} - 1), 0)$
$\alpha_{i,M} = \frac{1}{N}$	$u_{i,M} = \max(1 - f_{i,M}, 0)$ $v_{i,M} = 0$	- -

Fig. 16. The optimality conditions of NOC-SVM. (Upper: $m = 1, 2, \ldots, M-1$, and Lower: $m = M$.) Empty entries indicate cases that cannot occur.

[22] M. Bazaraa, H. Sherali, and C. Shetty, *Nonliear Programming: Theory and Algorithms*, 3rd ed. Hoboken, NJ, USA: John Wiley & Sons, 2006.

[23] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. on Neural Networks*, vol. 12, pp. 181–201, 2001.

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

[25] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics*, pp. 1:80–83, 1945.

[26] G. Lee and C. Scott, "Nested support vector machines," Tech. Rep., 2008, http://www.eecs.umich.edu/~cscott.

[27] I. Steinwart, "How to compare different loss functions and their risks," *Constructive Approximation*, vol. 26, pp. 225–287, 2007.

Gyemin Lee received the B.S. in Electrical Engineering from Seoul National University, Seoul, Korea in 2001 and M.S. in Electrical Engineering from University of Michigan, Ann Arbor, MI in 2007, where he is currently pursuing Ph.D. in Electrical Engineering under the supervision of Prof. C. Scott. His research interests include machine learning, optimization, and statistical file matching.

Clayton Scott received the A.B. in Mathematics from Harvard University in 1998, and the M.S. and Ph.D. 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 theory and applications.