Nested Support Vector Machines

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

Abstract

The 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 a nesting constraint is desirable for applications requiring the simultaneous estimation of multiple sets, including clustering, anomaly detection, and ranking problems. We propose new quadratic programs whose solutions give rise to nested extensions of the one-class and cost-sensitive SVMs. Furthermore, like conventional SVMs, the solution paths in our construction are piecewise linear in the control parameters, with significantly fewer breakpoints. We also describe decomposition algorithms to solve the quadratic programs. These methods are compared to conventional SVMs on synthetic and benchmark data sets, and are shown to exhibit more stable rankings and decreased sensitivity to parameter settings.

Index Terms

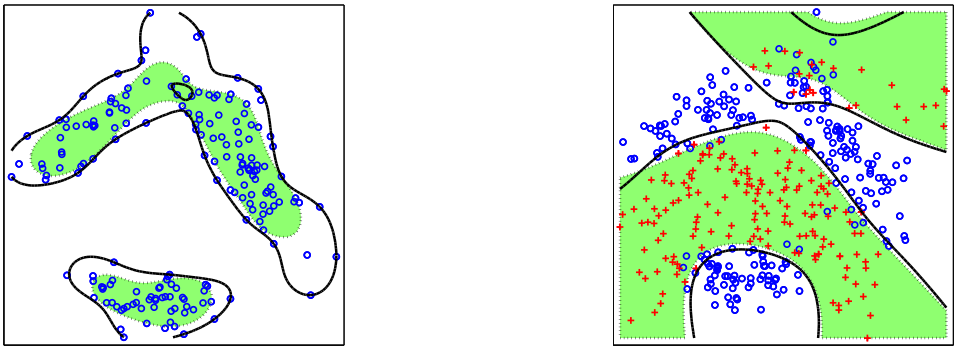
pattern classification, one class support vector machine, cost sensitive support vector machine, nested set estimation, solution paths.

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

G. Lee and C. Scott is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, 48103 USA e-mail: {gyemin, cscott}@eecs.umich.edu



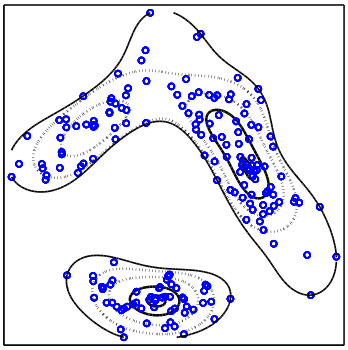
(a) OC-SVM (b) CS-SVM

Fig. 1. Two decision boundaries from the OC-SVM (a) and CS-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.

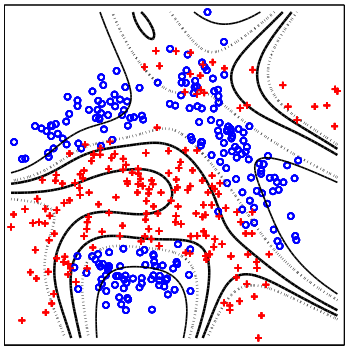
for many problems including clustering [1], outlier ranking [2], minimum volume set estimation [3], and anomaly detection [4]. Estimating cost-sensitive classifiers at a range of different cost asymmetries is important for ranking [5], Neyman-Pearson classification [6], transductive anomaly detection [7], and ROC studies [8].

Support vector machines (SVMs) are powerful nonparametric approaches to set estimation [9]. However, both the one-class SVM for level set estimation and the standard two-class SVM for classification do not produce set estimates that are *nested* as the parameter controlling the density level or, respectively, misclassification cost 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 in these two applications, 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].

In this paper, we develop nested variants of the one-class and two-class SVMs by incorporating nesting constraints into the dual quadratic programs defining these methods. Decomposition algorithms for solving the modified duals are also presented. Like the solution paths for the conventional SVMs [10], [8], [11], the nested SVM solution paths are also piecewise linear in the control parameters, but require far fewer



(a) Nested OC-SVM



(b) Nested CS-SVM

Fig. 2. Five decision boundaries from the Nested OC-SVM (a) and Nested CS-SVM (b) at five different density levels and cost asymmetries, respectively. These decision boundaries from the 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.

breakpoints. We compare our nested paths to the unnested paths on synthetic and benchmark data sets. We also quantify the degree to which standard SVMs are unnested, which is often quite high.

A. Motivating Applications

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

Ranking : In the bipartite ranking problem [12], we are given labeled examples from two classes, and the goal is to construct a score function that rates new examples according to their likelihood of belonging to the positive class. Cost-Sensitive SVMs (CS-SVMs) can be applied to this problem by varying the cost parameter, but the resulting sets are not nested, and therefore produce ambiguous rankings. Similarly, One-Class SVMs (OC-SVMs) can be applied to ranking the examples in an unlabeled dataset from an unknown density. In both cases, nested SVMs will make the score functions unambiguous and less sensitive to perturbations of the data. See Section V-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" [13] that depicts the bifurcation of clusters into disjoint components and gives a hierarchical representation of cluster structure. Therefore, a cluster tree can be estimated by training a OC-SVM at all values of the density level parameter.

Anomaly Detection : Anomaly detection aims to identify deviations from the normal data when combined observations of normal 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 the SVM.

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 in and $y_i \in \{-1, +1\}$ is its class. Conceptually, the SVM builds decision sets in two steps. First, each data point is mapped via a nonlinear map $\mathbf{x} \mapsto \Phi(\mathbf{x})$ into a high dimensional space $\mathcal{H}$ generated by a positive definite kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. This kernel corresponds to an inner product in $\mathcal{H}$ through $k(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle$. Second, the mapped data points in $\mathcal{H}$ are separated by a hyperplane with maximum margin (the distance between the hyperplane and its closest data point). If $\mathbf{w}$ and b denote the normal vector and the offset, then the two half-spaces of the hyperplane $\{\Phi(\mathbf{x}) : f(\mathbf{x}) \equiv \langle \mathbf{w}, \Phi(\mathbf{x}) \rangle + b = 0\}$ form positive and negative decision sets. If data are not linearly separable in $\mathcal{H}$, data points are allowed to be on the other side of the soft margin ($f(\mathbf{x}) = \pm 1$) through non-negative slack variables ξ_i. This can be achieved by solving

$$\min_{\mathbf{w}, \boldsymbol{\xi}} \frac{\lambda}{2} \|\mathbf{w}\|^2 + \sum_i \xi_i$$

$$\text{s.t.} \quad y_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad \text{for} \ \ \forall i$$

where λ controls regularization. In this formulation, the offset b is often omitted when Gaussian or inhomogeneous polynomial kernels are chosen [14]. More detailed discussion on the SVM can be found in [9].

A. Cost-Sensitive SVM

The SVM above, which we call the cost-insensitive SVM (CI-SVM) without offset, 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 [15].

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

$$\min_{\mathbf{w},\boldsymbol{\xi}} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \gamma \sum_{I_+} \xi_i + (1-\gamma) \sum_{I_-} \xi_i \tag{1}$$

$$\text{s.t.} \quad y_i \langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad \text{for} \ \ \forall i$$

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

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

$$\min_{\boldsymbol{\alpha}} \frac{1}{2\lambda} \sum_i \sum_j \alpha_i \alpha_j y_i y_j K_{i,j} - \sum_i \alpha_i \tag{2}$$

$$\text{s.t.} \quad 0 \leq \alpha_i \leq \mathbf{1}_{\{y_i < 0\}} + y_i \gamma \quad \text{for} \ \ \forall i.$$

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.

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 $\mathbf{x}$. This decision function takes only non-positive values when $\gamma = 0$, and corresponds to $(0, 0)$ in the ROC curve. On the other hand, $\gamma = 1$ penalizes only the violations of positive examples, and corresponds to $(1, 1)$ in the ROC curve.

Bach et al. [8] extended the method of Hastie et al. [10] 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 [16], [17] 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, typically the negative class, $y_i = -1, \forall i$. The primal quadratic program of the OC-SVM is

$$\min_{\mathbf{w},\boldsymbol{\xi}} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{N} \sum_{i=1}^{N} \xi_i \tag{4}$$

$$\text{s.t.} \quad \langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad \text{for} \ \ \forall i.$$

This problem is again solved via its dual in practice:

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

Then 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 shown piecewise linear in λ [11]. 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

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

Vert and Vert [18] showed that by modifying this estimate slightly, substituting $\alpha_i^*(\eta\lambda)$ for $\alpha_i^*(\lambda)$ where $\eta > 1$, leads to a consistent estimate of the true level set. Regardless of whether $\eta = 1$ or $\eta > 1$, however, the obtained estimates are not guaranteed to be nested as we will see in Section V. Note also that when $\alpha_i^*(\lambda) = \frac{1}{N}$, (6) becomes equivalent to set estimation based on kernel density estimation.

III. Nested CS-SVM

In this section, we develop a nested cost-sensitive 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 the 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 Nested CS-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 $0 = \gamma_1 < \gamma_2 < \cdots < \gamma_M = 1$, the Nested CS-SVM

solves

$$\min_{\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_M} \sum_{m=1}^{M} \left[\frac{1}{2\lambda} \sum_i \sum_j \alpha_{i,m}\alpha_{j,m} y_i y_j K_{i,j} - \sum_i \alpha_{i,m} \right] \tag{7}$$

$$\text{s.t.} \quad 0 \le \alpha_{i,m} \le \mathbf{1}_{\{y_i<0\}} + y_i\gamma_m \quad \text{for} \ \ \forall i, m \tag{8}$$

$$y_i\alpha_{i,1} \le y_i\alpha_{i,2} \le \cdots \le y_i\alpha_{i,M} \quad \text{for} \ \ \forall i \tag{9}$$

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 VI, the proposed quadratic program for the Nested OC-SVM 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:

$$\alpha^*_i(\gamma) = \epsilon\alpha^*_{i,1} + (1-\epsilon)\alpha^*_{i,2}. \tag{10}$$

This is motivated by the piecewise linearity of the Lagrange multipliers of the CS-SVM. Then the positive decision set at cost asymmetry γ is

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

Proposition 1. *The Nested CS-SVM equipped with a kernel such that $k(\cdot,\cdot) \ge 0$ (e.g., Gaussian kernels or polynomial kernels of even orders) generates nested decision sets. In other words, if $0 \le \gamma_\epsilon < \gamma_\delta \le 1$, 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 Nested CS-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 $k(\cdot,\cdot) \ge 0$ and (9), we have $\sum_i \alpha^*_{i,1} y_i k(\mathbf{x}_i, \mathbf{x}) \le \sum_i \alpha^*_{i,2} y_i k(\mathbf{x}_i, \mathbf{x})$. 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 $0 \leq \gamma_\epsilon < \gamma_\delta \leq 1$. 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, $\widehat{G}_{\gamma_\epsilon} \subset \widehat{G}_{\gamma_\delta}$. ■

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 [19], 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 [20], the algorithm will converge in far fewer iterations. In the Appendix, we further discuss the data point selection scheme and termination criterion based on the KKT optimality condition.

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 $\alpha_{1,m}$:

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

where $\alpha_{1,m}^{\text{old}}$ and $f_{1,m} = \frac{1}{\lambda}\left(\sum_{j\neq 1}\alpha_{j,m}y_jK_{1,j} + \alpha_{1,m}^{\text{old}}y_1K_{1,1}\right)$ denote the variable and corresponding output before 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,

$$\min_{\alpha_{1,1},\ldots,\alpha_{1,M}} \sum_m \left[\frac{1}{2}\alpha_{1,m}^2 - \alpha_{1,m}\alpha_{1,m}^{\text{new}}\right] \tag{12}$$

$$\text{s.t.} \quad 0 \leq \alpha_{1,m} \leq \mathbf{1}_{\{y_1<0\}} + y_1\gamma_m \quad \text{for} \quad \forall m \tag{13}$$

$$y_1\alpha_{1,1} \leq y_1\alpha_{1,2} \leq \cdots \leq y_1\alpha_{1,M} \tag{14}$$

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 via standard quadratic program solvers.

IV. NESTED OC-SVM

In this section, we present a nested extension of the OC-SVM. The Nested OC-SVM estimates a family of nested level sets over a continuum of levels λ. Our approach here parallels the approach developed for the CS-SVM. First, we will introduce an objective function for nested set estimation, and will develop analogous interpolation and decomposition algorithms for the Nested OC-SVM.

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

Initialize:

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

repeat

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

Compute:

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

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

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

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

$$\text{s.t.} \quad 0 \le \alpha_{i,m} \le \mathbf{1}_{\{y_i<0\}} + y_i\gamma_m \quad \text{for} \quad \forall m$$

$$y_i\alpha_{i,1} \le y_i\alpha_{i,2} \le \cdots \le y_i\alpha_{i,M}$$

until Accuracy conditions are satisfied

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

Fig. 3. Decomposition algorithm for the Nested Cost-Sensitive Support Vector Machine. Specific strategies for data point selection and termination, based on the KKT conditions, are given in the Appendix.

A. Finite Family of Nested Sets

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

$$\min_{\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_M} \sum_{m=1}^M \left[\frac{1}{2\lambda_m}\sum_i\sum_j \alpha_{i,m}\alpha_{j,m}K_{i,j} - \sum_i \alpha_{i,m}\right] \tag{15}$$

$$\text{s.t.} \quad 0 \le \alpha_{i,m} \le \frac{1}{N} \quad \text{for} \quad \forall i, m \tag{16}$$

$$\frac{\alpha_{i,1}}{\lambda_1} \le \frac{\alpha_{i,2}}{\lambda_2} \le \cdots \le \frac{\alpha_{i,M}}{\lambda_M} \quad \text{for} \quad \forall i \tag{17}$$

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 V-B, Appendix C). In Section VI, the proposed quadratic program for the Nested OC-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

$$\alpha^*_i(\lambda) = \epsilon\alpha^*_{i,1} + (1-\epsilon)\alpha^*_{i,2}.$$

It is motivated by the piecewise linearity of the OC-SVM solutions in λ. For $\lambda > \lambda_1$ or $\lambda < \lambda_M$, we set $\alpha^*_i(\lambda) = \alpha^*_{i,1}$ or $\alpha^*_i(\lambda) = \alpha^*_{i,M}$ for $\forall i$, respectively. Then the level set estimate becomes

$$\widehat{G}_\lambda = \{\mathbf{x} : \sum_i \alpha^*_i(\lambda)k(\mathbf{x}_i,\mathbf{x}) > \lambda\}. \tag{18}$$

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

Proposition 2. *The Nested OC-SVM with a Gaussian kernel generates nested density level set estimates. Therefore, 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 (15) satisfy $\widehat{G}_{\lambda_1} \subset \widehat{G}_{\lambda_2} \subset \cdots \subset \widehat{G}_{\lambda_M}$. Second, the interpolated set (18) 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 Nested OC-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 (17), 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 (17), 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

$$\sum_i (\delta\alpha^*_{i,1} + (1-\delta)\alpha^*_{i,2})k(\mathbf{x}_i, \mathbf{x}) > \lambda_\delta \tag{19}$$

$$\sum_i \alpha^*_{i,2} k(\mathbf{x}_i, \mathbf{x}) > \lambda_2. \tag{20}$$

By $\frac{\epsilon}{\delta}\times$ (19) + $(1-\frac{\epsilon}{\delta})\times$ (20), we have $\sum_i (\epsilon\alpha^*_{i,1} + (1-\epsilon)\alpha^*_{i,2})k(\mathbf{x}_i, \mathbf{x}) > \lambda_\epsilon$. Thus, $\widehat{G}_{\lambda_\delta} \subset \widehat{G}_{\lambda_\epsilon}$. ■

C. Decomposition Algorithm

We also use a decomposition algorithm to solve (15). The general steps are the same as explained in Section III-C for the Nested CS-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

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

where $\alpha^{\text{old}}_{1,m}$ and $f_{1,m} = \frac{1}{\lambda_m}\left(\sum_{j\neq 1}\alpha_{j,m}K_{1,j} + \alpha^{\text{old}}_{1,m}K_{1,1}\right)$ denote the variable from the previous iteration step and the corresponding output, respectively. C is a constant that does not affect the solution.

Then we obtain the reduced optimization problem of M variables,

$$\min_{\alpha_{1,1},\ldots,\alpha_{1,M}} \sum_m \left[\frac{1}{2\lambda_m}\alpha^2_{1,m} - \frac{\alpha_{1,m}}{\lambda_m}\alpha^{\text{new}}_{1,m} \right] \tag{21}$$

$$\text{s.t.} \quad 0 \le \alpha_{1,m} \le \frac{1}{N} \quad \text{for} \quad \forall m \tag{22}$$

$$\frac{\alpha_{1,1}}{\lambda_1} \le \frac{\alpha_{1,2}}{\lambda_2} \le \cdots \le \frac{\alpha_{1,M}}{\lambda_M} \tag{23}$$

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

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

Initialize:

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

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:

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

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

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

until Accuracy conditions are satisfied

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

Fig. 4. Decomposition algorithm for the Nested One-Class Support Vector Machine.

V. Experiments and Results

In order to compare the algorithms described above, we experimented on 13 benchmark data sets available online [1]. Their brief summary is provided in Fig. 5. Each feature is standardized with zero mean and unit variance. The first eleven data sets are randomly permuted 100 times and divided into training and test sets, and the last two data sets are permuted 20 times. 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.

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

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

Each algorithm generates a family of decision functions and set estimates. From these sets, we construct a ROC curve 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 the CS-SVM or the OC-SVM are not properly nested, and cause ambiguity particularly in ranking applications. In Section V-C, we measure this violation of the nesting by defining the *ranking disagreement* of two rank scoring functions. Then in Section V-D, we combine this ranking disagreement and the AUC, and compare the algorithms over multiple data sets using the Wilcoxon signed ranks test and the Friedman test as suggested in [21].

A. Two-class Problems

The CI-SVM, CS-SVM, and Nested CS-SVM are compared in two-class problems. For the Nested CS-SVM, we set $M = 5$ and solved (7) at cost asymmetries $\gamma = (0, 0.25, 0.50, 0.75, 1)$.

In two-class problems, we also searched for the regularization parameter λ over 10 logarithmically space 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).

Data Set	CI	CS	NCS
banana	0.9598 (± 0.0028)	0.9505 (± 0.0098)	0.9630 (± 0.0039)
breast-cancer	0.7029 (± 0.0553)	0.7332 (± 0.0543)	0.7315 (± 0.0562)
diabetes	0.8298 (± 0.0160)	0.8291 (± 0.0169)	0.8258 (± 0.0170)
flare-solar	0.6481 (± 0.0339)	0.6586 (± 0.0404)	0.5802 (± 0.0476)
german	0.7885 (± 0.0250)	0.7963 (± 0.0245)	0.7888 (± 0.0245)
heart	0.9067 (± 0.0253)	0.9088 (± 0.0272)	0.9074 (± 0.0277)
ringnorm	0.9988 (± 0.0001)	0.9825 (± 0.0029)	0.9556 (± 0.0119)
thyroid	0.9897 (± 0.0093)	0.9620 (± 0.0370)	0.9541 (± 0.0378)
titanic	0.6102 (± 0.0693)	0.5990 (± 0.0698)	0.5978 (± 0.0708)
twonorm	0.9977 (± 0.0001)	0.9977 (± 0.0001)	0.9977 (± 0.0003)
waveform	0.9661 (± 0.0035)	0.9699 (± 0.0023)	0.9677 (± 0.0031)
image	0.9930 (± 0.0022)	0.9918 (± 0.0023)	0.9858 (± 0.0042)
splice	0.9578 (± 0.0036)	0.9508 (± 0.0039)	0.9519 (± 0.0045)

Fig. 6. Comparison of AUC between the Cost-Insensitive (CI), the Cost-Sensitive (CS), and the Nested CS-SVM.

We compared the described algorithms by constructing ROC curves and computing their AUC. Since the CI-SVM only corresponds to a single point on the ROC curve, we shifted its offset from $-\infty$ to ∞ and extended to a whole ROC curve. The results are collected in Fig. 6.

B. One-class Problems

For the Nested OC-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 dataset, we chose the negative examples for training. Since the second class was unavailable at training time, we simulated an artificial second class from a uniform distribution. For evaluation of the trained decision functions, both the positive samples in the test sets and the uniform samples were used as the alternative class. Fig. 7 reports the experiment results for both cases (denoted by Positive and Uniform, respectively).

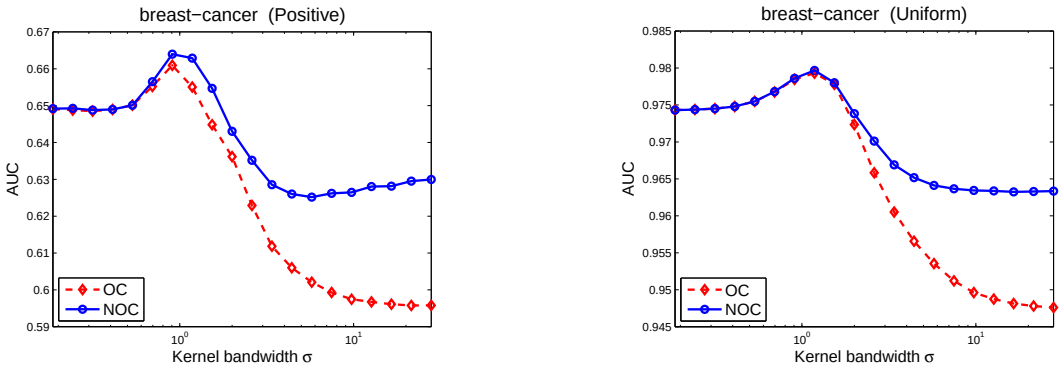
Fig. 8 shows the AUC of the two algorithms over a range of σ. Throughout the experiments on one-class problems, we observed that the nested OC-SVM is more robust to the kernel bandwidth selection than the OC-SVM. However, we did not observe the similar results on two-class problems.

C. Ranking disagreement

The decision sets from the OC-SVM and the CS-SVM are not properly nested. Fig. 1 illustrates examples of two set estimates violating nesting condition. Since larger λ means higher density level,

	Positive		Uniform	
Data Set	OC	NOC	OC	NOC
banana	0.9192 (± 0.0096)	0.9300 (± 0.0072)	0.9065 (± 0.0037)	0.9114 (± 0.0031)
breast-cancer	0.6477 (± 0.0621)	0.6545 (± 0.0617)	0.9763 (± 0.0067)	0.9766 (± 0.0067)
diabetes	0.7224 (± 0.0237)	0.7324 (± 0.0229)	0.9961 (± 0.0019)	0.9961 (± 0.0019)
flare-solar	0.6017 (± 0.0429)	0.6014 (± 0.0438)	0.9985 (± 0.0006)	0.9984 (± 0.0006)
german	0.6268 (± 0.0313)	0.6268 (± 0.0312)	0.9911 (± 0.0032)	0.9911 (± 0.0032)
heart	0.7767 (± 0.0373)	0.7828 (± 0.0367)	0.9861 (± 0.0050)	0.9863 (± 0.0051)
ringnorm	0.9970 (± 0.0003)	0.9970 (± 0.0003)	1.0000 (± 0.0000)	1.0000 (± 0.0000)
thyroid	0.9867 (± 0.0090)	0.9871 (± 0.0084)	0.9999 (± 0.0001)	0.9999 (± 0.0001)
titanic	0.6021 (± 0.0686)	0.5883 (± 0.0627)	0.7619 (± 0.0514)	0.7653 (± 0.0419)
twonorm	0.9105 (± 0.0116)	0.9126 (± 0.0099)	1.0000 (± 0.0000)	1.0000 (± 0.0000)
waveform	0.7520 (± 0.0192)	0.7622 (± 0.0171)	1.0000 (± 0.0000)	1.0000 (± 0.0000)
image	0.8720 (± 0.0392)	0.8542 (± 0.0393)	1.0000 (± 0.0000)	1.0000 (± 0.0000)
splice	0.4165 (± 0.0095)	0.4158 (± 0.0095)	0.5537 (± 0.0121)	0.5545 (± 0.0080)

Fig. 7. Comparison of AUC of algorithms on one-class problem. Unnested One-Class SVM and Nested OC-SVM are compared. Left columns are the cases when the alternative hypotheses are from the positive class samples in the data sets, and right columns are when the alternative hypotheses are from uniform distributions.



(a) Positive (b) Uniform

Fig. 8. The effect of kernel bandwidth σ on the performance (AUC). The Nested OC-SVM is less sensitive to σ than the OC-SVM.

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 properly 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:

$$s_+(\mathbf{x}) = 1 - \min_{\{\gamma: f_\gamma(\mathbf{x}) \geq 0\}} \gamma, \qquad s_-(\mathbf{x}) = 1 - \max_{\{\gamma: f_\gamma(\mathbf{x}) \leq 0\}} \gamma. \tag{24}$$

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

$$s_+(\mathbf{x}) = \max_{\{\lambda: \mathbf{x} \in \widehat{G}_\lambda\}} \lambda, \qquad s_-(\mathbf{x}) = \min_{\{\lambda: \mathbf{x} \in \widehat{G}_\lambda\}} \lambda. \tag{25}$$

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*:

$$d(s_+, s_-) = \frac{1}{N} \sum_i \max_{j \neq i} I_{\{(s_+(\mathbf{x}_i) - s_+(\mathbf{x}_j))(s_-(\mathbf{x}_i) - s_-(\mathbf{x}_j)) < 0\}}, \tag{26}$$

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 the OC-SVM. In the table, d_2 refers to the ranking disagreement of the CS-SVM, and d_1 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 difference between the above ranking functions.

D. Statistical comparison

We employ the statistical methodology of Demšar [21] to compare the algorithms across all data sets. For two-class problems, we use the Friedman test to compare the CI-SVM, CS-SVM, and Nested CS-SVM. For one-class problems, we use the Wilcoxon signed ranks test and compare the OC-SVM and the Nested OC-SVM.

The Friedman test is a non-parametric method for testing the significance of differences between more than two sample means. Algorithms are ranked for each data set, and their average ranks are compared (rank 1 for the best). Fig. 10 reports the average ranks of the three algorithms on the benchmark data sets

Data set	$d_2(s_+, s_-)$	$d_1(s_+, s_-)$	$d_u(s_+, s_-)$
banana	0.0243	0.4984	0.3896
breast-cancer	0.0134	0.2529	0.0933
diabetes	0.1193	0.0206	0.0012
flare-solar	0.3003	0.6572	0.1989
german	0.0191	0.0000	0.0000
heart	0.0053	0.0002	0.0000
ringnorm	0.2447	0.0000	0.0000
thyroid	0.0023	0.0191	0.0003
titanic	0.0000	0.2500	0.2310
twonorm	0.0068	0.0006	0.0005
waveform	0.0789	0.0021	0.0011
image	0.3079	0.2766	0.0479
splice	0.1055	0.0000	0.0000

Fig. 9. The measure of disagreement of the two ranking functions from the CS-SVM and the OC-SVM. The meaning of each subscript is explained in the text. s_+ and s_- are defined in (24) and (25).

CI	CS	NCS	F_F
1.77	1.77	2.46	2.28

Fig. 10. Comparison of the AUCs of the three algorithms: Cost-Insensitive (CI), Cost-Sensitive (CS), and Nested CS-SVM (NCS) using the Friedman Test. The Friedman test compares the average ranks of algorithms. The critical difference of F_F is 3.40 at $\alpha = 0.05$. Therefore, no significant difference is detected.

along with a test statistic F_F. Under the null hypothesis of no significant differences among the algorithms, F_F is distributed approximately according to an F-distribution with $(3-1)$ and $(3-1)(13-1)$ degrees of freedom. The critical value of $F(2, 24)$ for a confidence level $\alpha = 0.05$ is 3.40, so a significant difference between the algorithms was not detected by the Friedman test.

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 *two algorithms* over multiple data sets. Fig. 11 reports the comparison results of the algorithms. The difference values between the AUCs from the two algorithms are ranked ignoring the signs, and then the ranks of positive (and respectively negative) differences are added. Here the numbers of right column denote the sums of ranks of the data sets on which the Nested OC-SVM performed better than the OC-SVM; the left column is for the opposite. T is the smaller of the two sums. For a confidence level of $\alpha = 0.05$ and 13 data sets, the difference between algorithms is significant if T is less than or equal to 17 [22]. Therefore, any

	OC	NOC	T
Positive	35	56	35
Uniform	21.5	69.5	21.5

Fig. 11. Comparison of Unnested and Nested OC-SVM using the Wilcoxon signed ranks test. In the evaluation of the unlabeled problems, both the cases of alternative hypothesis are accounted. Left (right) column is the sums of the ranks of the data sets on which the OC-SVM (Nested OC-SVM) outperforms the Nested OC-SVM (OC-SVM). T is the smaller of the two sums. For $\alpha = 0.05$, the difference is significant if $T \leq 17$.

CI	CS	NCS	F_F
1.46	2.77	1.77	10.53

	OC	NOC	T
Positive	5	86	5
Uniform	0	91	0

Fig. 12. Comparison of algorithms based on the AUC along with the disorder measure. Left: Friedman Test on Cost-Insensitive (CI), Cost-Sensitive (CS), and Nested CS-SVM (NCS). Right: Wilcoxon Signed-Ranks Test on Unnested and Nested OC-SVM.

significant performance difference between the OC-SVM and the Nested OC-SVM was not detected in the test.

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 Friedman test and the Wilcoxon signed-ranks test using this combined performance measure. From the results, we can observe clearly the performance differences between algorithms. The Friedman test shows that the performance difference exists between algorithms because F_F is greater than the critical value 3.40. After the Friedman test, we proceed with a post-hoc, Nemenyi test, to find which algorithms actually differ. The critical difference of the Nemenyi test is 0.92 for three algorithms and a confidence level $\alpha = 0.05$. If the average ranks differ more than this critical value, then the performance of two algorithms is significantly different. Thus, the Nested CS-SVM and the CI-SVM outperforms the CS-SVM. The performance difference between the OC-SVM and the Nested OC-SVM is also detected by the Wilcoxon test since the test statistic T is smaller than the critical difference 17 for both cases of the second class. Therefore, we can conclude that the nested algorithms perform better than their unnested counterparts.

VI. Connections and Interpretations

We can find a primal optimization problem of the Nested CS-SVM if we think (7) is a dual problem:

$$\begin{aligned}\min_{\{\mathbf{w}_m\},\{\xi_{i,m}\}} & \sum_{m=1}^{M}\left[\frac{\lambda}{2}\|\mathbf{w}_m\|^2+\gamma_m\sum_{I_+}\xi_{i,m}+(1-\gamma_m)\sum_{I_-}\xi_{i,m}\right] \\ \text{s.t.} \quad & \sum_{k=m}^{M}\langle\mathbf{w}_k,\Phi(\mathbf{x}_i)\rangle\geq\sum_{k=m}^{M}(1-\xi_{i,k}) \quad \text{for} \quad i\in I_+,\forall m \\ & \sum_{k=1}^{m}\langle\mathbf{w}_k,\Phi(\mathbf{x}_i)\rangle\leq-\sum_{k=1}^{m}(1-\xi_{i,k}) \quad \text{for} \quad i\in I_-,\forall m \\ & \xi_{i,m}\geq 0 \quad \text{for} \quad \forall i,m.\end{aligned} \tag{27}$$

In the appendix, we show that this is indeed the primal by deriving (7) from (27). Note that (27) reduces to the primal of the CS-SVM (1) for $M=1$.

Likewise, we can also derive the primal form of the Nested OC-SVM:

$$\begin{aligned}\min_{\{\mathbf{w}_m\},\{\xi_{i,m}\}} & \sum_{m=1}^{M}\left[\frac{\lambda_m}{2}\|\mathbf{w}_m\|^2+\frac{1}{N}\sum_{i}\xi_{i,m}\right] \\ \text{s.t.} \quad & \sum_{k=m}^{M}\lambda_k\langle\mathbf{w}_k,\Phi(\mathbf{x}_i)\rangle\geq\sum_{k=m}^{M}\lambda_k(1-\xi_{i,m}) \quad \text{for} \quad \forall i,m \\ & \xi_{i,m}\geq 0 \quad \text{for} \quad \forall i,m,\end{aligned} \tag{28}$$

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

With these formulations, we can see the geometric meaning of $\mathbf{w}$ and $\boldsymbol{\xi}$. For simplicity, consider (28) when $M=2$:

$$\begin{aligned}\min_{\{\mathbf{w}_m\},\{\xi_{i,m}\}} & \frac{\lambda_2}{2}\|\mathbf{w}_2\|^2+\frac{1}{N}\sum_{i}\xi_{i,2}+\frac{\lambda_1}{2}\|\mathbf{w}_1\|^2+\frac{1}{N}\sum_{i}\xi_{i,1} \\ \text{s.t.} \quad & \langle\lambda_2\mathbf{w}_2,\Phi(\mathbf{x}_i)\rangle\geq\lambda_2(1-\xi_{i,2}) \quad \text{for} \quad \forall i \\ & \langle\lambda_2\mathbf{w}_2+\lambda_1\mathbf{w}_1,\Phi(\mathbf{x}_i)\rangle\geq\lambda_2(1-\xi_{i,2})+\lambda_1(1-\xi_{i,1}) \quad \text{for} \quad \forall i \\ & \xi_{i,m}\geq 0 \quad \text{for} \quad \forall i,m.\end{aligned} \tag{29}$$

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 $P_{\frac{\lambda_2\mathbf{w}_2+\lambda_1\mathbf{w}_1}{\lambda_2+\lambda_1}}$.

VII. Conclusion

In this paper, we introduced a novel framework for building a family of nested sets for the tasks of density level set estimation or cost-sensitive classification. Our approach is based on the large margin principle and does not rely on direct density estimation. The key step involves forming new quadratic programs with constraints imposing nesting structure. Our construction generates a finite number of nested set estimates at a set of preselected parameters, and linearly interpolates these sets to a continuous nested family. We also developed efficient algorithms to solve the proposed quadratic problems. Therefore, the Nested OC-SVM yields a family of nested density level set estimates indexed by density level λ, and the Nested CS-SVM yields a family of nested classifiers indexed by cost asymmetry γ. These results sharply contrast to the outputs of the original SVMs that are not nested. Hence, the usefulness of the nested SVMs is obvious because their outcomes can be readily applied to many applications requiring multiple set estimations including clustering, ranking, and anomaly detection.

We also investigated the CS-SVM and OC-SVM for ranking problems. Ranking functions driven by the CS-SVM and OC-SVM can be benefited from non-parallel directions in kernel feature space unlike previous approaches employing a fixed direction with a varying offset. We demonstrated that the ranking score functions from the original SVMs can cause ranking disagreements, while their nested extensions generate consistent ranking functions.

Appendix A
Data Point Selection and Termination Condition of the Nested CS-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.

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

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

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

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \alpha_{i,m}} &= y_i f_{i,m} - 1 + u_{i,m} \begin{cases} -v_{i,m} + v_{i,m+1} & \text{for} \quad i \in I_+ \\ +v_{i,m-1} - v_{i,m} & \text{for} \quad i \in I_- \end{cases} \\ &= 0 \end{aligned} \tag{30}$$

where, recall, $f_{i,m} = \frac{1}{\lambda} \sum_j \alpha_{j,m} y_j K_{i,j}$ 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

$$y_i f_{i,m} - 1 + u_{i,m} = \begin{cases} v_{i,m} - v_{i,m+1} & \text{for} \quad i \in I_+ \\ -v_{i,m-1} + v_{i,m} & \text{for} \quad i \in I_- \end{cases} \tag{31}$$

$$0 \leq \alpha_{i,m} \leq \mathbf{1}_{\{y_i < 0\}} + y_i \gamma_m \qquad \text{for } \forall i, \forall m \tag{32}$$

$$y_i \alpha_{i,1} \leq y_i \alpha_{i,2} \leq \cdots \leq y_i \alpha_{i,M} \qquad \text{for } \forall i \tag{33}$$

$$u_{i,m} \left(\alpha_{i,m} - \mathbf{1}_{\{y_i < 0\}} - y_i \gamma_m \right) = 0 \qquad \text{for } \forall i, \forall m \tag{34}$$

$$v_{i,m}(\alpha_{i,m-1} - \alpha_{i,m}) = 0 \qquad \text{for } i \in I_+, \forall m \tag{35}$$

$$v_{i,m}(\alpha_{i,m} - \alpha_{i,m+1}) = 0 \qquad \text{for } i \in I_-, \forall m \tag{36}$$

$$u_{i,m} \geq 0, \qquad v_{i,m} \geq 0 \qquad \text{for } \forall i, m. \tag{37}$$

Since (7) is a convex program, the KKT conditions are also sufficient [20]. That is, $\alpha_{i,m}$, $u_{i,m}$, and $v_{i,m}$ satisfying (31)-(37) 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

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

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 (30)), whereas $f_{i,m}$ can be easily computed from the known variables $\alpha_{i,m}$. Fig. 13 and Fig. 14 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

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

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

Fig. 13. The optimality conditions of the Nested CS-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 the 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. 14. The optimality conditions of the Nested CS-SVM when $i \in I_-$. (Upper: $m = 2, \ldots, M$, Lower: $m = 1$.)

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. 13 iteratively from $m = M$ down to $m = 1$. If $i \in I_-$, we use Fig. 14 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 THE NESTED OC-SVM

As in the Nested CS-SVM, we investigate the optimality condition of the Nested OC-SVM (15) and find a data point selection method and a termination condition.

With a slight modification, we rewrite (15),

$$\min_{\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_M} \sum_{m=1}^{M} \left[\frac{1}{2\lambda_m} \sum_i \sum_j \alpha_{i,m}\alpha_{j,m}K_{i,j} - \sum_i \alpha_{i,m} \right] \tag{38}$$

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

$$0 \le \frac{\alpha_{i,1}}{\lambda_1} \le \frac{\alpha_{i,2}}{\lambda_2} \le \cdots \le \frac{\alpha_{i,M}}{\lambda_M} \quad \text{for} \quad \forall i.$$

We will use the KKT conditions to find the necessary and sufficient conditions of the optimal solution of above. The Lagrangian is

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

$$- \sum_i v_{i,1} \frac{\alpha_{i,1}}{\lambda_1} + \sum_i \sum_{m=2}^{M} v_{i,m} \left(\frac{\alpha_{i,m-1}}{\lambda_{m-1}} - \frac{\alpha_{i,m}}{\lambda_m} \right)$$

where $u_{i,m} \ge 0$ and $v_{i,m} \ge 0$ for $\forall i, m$. The derivative of $\mathcal{L}$ with respect to $\alpha_{i,m}$ vanishes

$$\frac{\partial \mathcal{L}}{\partial \alpha_{i,m}} = f_{i,m} - 1 + u_{i,m} \begin{cases} -\frac{v_{i,m}}{\lambda_m} + \frac{v_{i,m+1}}{\lambda_m} & \text{for} \quad m = 1, \ldots, M-1 \\ -\frac{v_{i,M}}{\lambda_M} & \text{for} \quad m = M \end{cases} \tag{39}$$

$$= 0. \tag{40}$$

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

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

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

Fig. 15. The optimality conditions of the Nested OC-SVM. (Upper: $m = 1, 2, \ldots, M-1$, and Lower: $m = M$.)

constraints for $\mathbf{x}_i$:

$$f_{i,m} - 1 + u_{i,m} = \begin{cases} \frac{v_{i,m}}{\lambda_m} - \frac{v_{i,m+1}}{\lambda_m} & \text{for} \quad m = 1, \ldots, M-1 \\ \frac{v_{i,M}}{\lambda_M} & \text{for} \quad m = M \end{cases} \tag{41}$$

$$\alpha_{i,m} \leq \frac{1}{N} \qquad \text{for } \forall m \tag{42}$$

$$0 \leq \frac{\alpha_{i,1}}{\lambda_1} \leq \frac{\alpha_{i,2}}{\lambda_2} \leq \cdots \leq \frac{\alpha_{i,M}}{\lambda_M} \tag{43}$$

$$u_{i,m}(\alpha_{i,m} - \frac{1}{N}) = 0 \qquad \text{for } \forall m \tag{44}$$

$$v_{i,m}(\frac{\alpha_{i,m-1}}{\lambda_{m-1}} - \frac{\alpha_{i,m}}{\lambda_m}) = 0 \qquad \text{for } \forall m \tag{45}$$

$$u_{i,m} \geq 0, \qquad v_{i,m} \geq 0 \qquad \text{for } \forall m. \tag{46}$$

Since (38) is a convex program, the KKT conditions are sufficient [20]. Therefore, $\alpha_{i,m}$, $u_{i,m}$, and $v_{i,m}$ satisfying (41)-(46) optimizes the objective function. On each round, the algorithm (Fig. 4) examines a current solution with these conditions and determines whether or not to stop. We measure the amount of error for $\mathbf{x}_i$ by defining

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

If the sum $\sum_i e_i$ decreases below a predetermined tolerance, the algorithm stops and returns the current solution. If not, the algorithm chooses the data point having the largest e_i and continues the iteration.

Computing e_i involves unknown variables $u_{i,m}$ and $v_{i,m}$ (see (39)), whereas $f_{i,m}$ can be computed with ease. In order to infer their values, we firstly assume the current solution $\alpha_{i,m}$ is optimal and secondly let $u_{i,m}$ and $v_{i,m}$ as in Fig. 15. The second step can be done by computing iteratively from $m = M$ 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 THE CS-SVM AND OC-SVM

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

First, we consider the OC-SVM. The decision function of the 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, denote the maximum row sum of the kernel matrix 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 in (1) of the CS-SVM. The decision function of the 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 the 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 $\lambda > \max\left(\lambda_{CS}(0), \lambda_{CS}(1)\right)$ generate the same solutions in the CS-SVM.

APPENDIX D
DERIVATION OF THE DUAL OPTIMIZATION PROBLEM OF THE NESTED CS-SVM

Here we will show that (27) is the primal of the quadratic program (7) for the Nested CS-SVM. First, we develop the Lagrangian with dual variables $u_{i,m} \geq 0$ and $v_{i,m} \geq 0$:

$$\begin{aligned}
\mathcal{L}_p(\mathbf{w}, \boldsymbol{\xi}, u, v) &= \sum_{m=1}^{M} \left[\frac{\lambda}{2} \|\mathbf{w}_m\|^2 + \gamma_m \sum_{i \in I_+} \xi_{i,m} + (1-\gamma_m) \sum_{i \in I_-} \xi_{i,m} \right] - \sum_{m=1}^{M} \sum_{i} v_{i,m} \xi_{i,m} \\
&\quad - \sum_{m=1}^{M} \sum_{i \in I_+} \sum_{k=m}^{M} u_{i,m} \left(\langle \mathbf{w}_k, \Phi(\mathbf{x}_i) \rangle - (1 - \xi_{i,k}) \right) \\
&\quad + \sum_{m=1}^{M} \sum_{i \in I_-} \sum_{k=1}^{m} u_{i,m} \left(\langle \mathbf{w}_k, \Phi(\mathbf{x}_i) \rangle + (1 - \xi_{i,k}) \right) \\
&= \sum_{m=1}^{M} \left[\frac{\lambda}{2} \|\mathbf{w}_m\|^2 + \gamma_m \sum_{i \in I_+} \xi_{i,m} + (1-\gamma_m) \sum_{i \in I_-} \xi_{i,m} \right] - \sum_{m=1}^{M} \sum_{i} v_{i,m} \xi_{i,m} \\
&\quad - \sum_{m=1}^{M} \sum_{i \in I_+} \sum_{k=1}^{m} u_{i,k} \left(\langle \mathbf{w}_m, \Phi(\mathbf{x}_i) \rangle - (1 - \xi_{i,m}) \right) \\
&\quad + \sum_{m=1}^{M} \sum_{i \in I_-} \sum_{k=m}^{M} u_{i,k} \left(\langle \mathbf{w}_m, \Phi(\mathbf{x}_i) \rangle + (1 - \xi_{i,m}) \right) \\
&= \sum_{m=1}^{M} \left[\frac{\lambda}{2} \|\mathbf{w}_m\|^2 + \gamma_m \sum_{i \in I_+} \xi_{i,m} + (1-\gamma_m) \sum_{i \in I_-} \xi_{i,m} \right] - \sum_{m=1}^{M} \sum_{i} v_{i,m} \xi_{i,m} \\
&\quad - \sum_{m=1}^{M} \sum_{i} \alpha_{i,m} y_i \left(\langle \mathbf{w}_m, \Phi(\mathbf{x}_i) \rangle - y_i (1 - \xi_{i,m}) \right)
\end{aligned} \tag{47}$$

where the last equality follows by letting

$$\alpha_{i,m} = \begin{cases} \sum_{k=1}^{m} u_{i,k} & \text{for} \quad i \in I_+ \\ \sum_{k=m}^{M} u_{i,k} & \text{for} \quad i \in I_-. \end{cases} \tag{48}$$

The derivatives with respect to $\mathbf{w}_m$ and $\xi_{i,m}$ vanish

$$\frac{\partial \mathcal{L}_p}{\partial \mathbf{w}_m} = \lambda \mathbf{w}_m - \sum_{i} \alpha_{i,m} y_i \Phi(\mathbf{x}_i) = 0 \tag{49}$$

$$\frac{\partial \mathcal{L}_p}{\partial \xi_{i,m}} = \mathbf{1}_{\{y_i < 0\}} + y_i \gamma_m - v_{i,m} - \alpha_{i,m} = 0. \tag{50}$$

Substituting these results (49), (50) into the above Lagrangian (47), we have

$$\begin{aligned}\mathcal{L}_p &= \sum_{m=1}^{M} \frac{\lambda}{2}\|\mathbf{w}_m\|^2 - \sum_{m=1}^{M}\sum_i \alpha_{i,m} y_i \langle \mathbf{w}_m, \Phi(\mathbf{x}_i)\rangle + \sum_{m=1}^{M}\sum_i \alpha_{i,m} \\ &= \sum_m \left[-\frac{1}{2\lambda}\sum_i\sum_j \alpha_{i,m}\alpha_{j,m} y_i y_j K_{i,j} + \sum_i \alpha_{i,m} \right].\end{aligned}$$

Combining the non-negativity of $u_{i,m}$ and $v_{i,m}$ to (48) and (50), the nesting constraints (9) and the box constraints (8) can be obtained.

APPENDIX E
DERIVATION OF THE DUAL OF THE NESTED OC-SVM

In this appendix, we show that (28) is the primal of the quadratic program (15) for the Nested OC-SVM. First, we develop the Lagrangian with dual variables $u_{i,m} \geq 0$ and $v_{i,m} \geq 0$:

$$\begin{aligned}\mathcal{L}_p(\mathbf{w}, \boldsymbol{\xi}, u, v) &= \sum_{m=1}^{M}\left[\frac{\lambda_m}{2}\|\mathbf{w}\|^2 + \frac{1}{N}\sum_i \xi_{i,m}\right] - \sum_{m=1}^{M}\sum_i v_{i,m}\xi_{i,m} \\ &\quad - \sum_{m=1}^{M}\sum_i\sum_{k=m}^{M} u_{i,m}\lambda_k\left(\langle \mathbf{w}_k, \Phi(\mathbf{x}_i)\rangle - (1-\xi_{i,k})\right) \\ &= \sum_{m=1}^{M}\left[\frac{\lambda_m}{2}\|\mathbf{w}\|^2 + \frac{1}{N}\sum_i \xi_{i,m}\right] - \sum_{m=1}^{M}\sum_i v_{i,m}\xi_{i,m} \\ &\quad - \sum_{m=1}^{M}\sum_i\sum_{k=1}^{m} u_{i,k}\lambda_m\left(\langle \mathbf{w}_m, \Phi(\mathbf{x}_i)\rangle - (1-\xi_{i,m})\right) \\ &= \sum_{m=1}^{M}\left[\frac{\lambda_m}{2}\|\mathbf{w}\|^2 + \frac{1}{N}\sum_i \xi_{i,m}\right] - \sum_{m=1}^{M}\sum_i v_{i,m}\xi_{i,m} \\ &\quad - \sum_{m=1}^{M}\sum_i \alpha_{i,m}\left(\langle \mathbf{w}_m, \Phi(\mathbf{x}_i)\rangle - (1-\xi_{i,m})\right)\end{aligned} \tag{51}$$

where the last equality follows by letting

$$\alpha_{i,m} = \lambda_m \sum_{k=1}^{m} u_{i,k}. \tag{52}$$

The derivatives with respect to $\mathbf{w}_m$ and $\xi_{i,m}$ vanish

$$\frac{\partial \mathcal{L}_p}{\partial \mathbf{w}_m} = \lambda_m \mathbf{w}_m - \sum_{m=1}^{M}\sum_i \alpha_{i,m}\Phi(\mathbf{x}_i) = 0 \tag{53}$$

$$\frac{\partial \mathcal{L}_p}{\partial \xi_{i,m}} = \frac{1}{N} - v_{i,m} - \alpha_{i,m} = 0. \tag{54}$$

Substituting these results (53), (54) in the above Lagrangian (51), we obtain

$$\begin{aligned}\mathcal{L}_p &= \sum_{m=1}^{M} \frac{\lambda_m}{2}\|\mathbf{w}_m\|^2 - \sum_{m=1}^{M}\sum_i \alpha_{i,m}\langle \mathbf{w}_m, \Phi(\mathbf{x}_i)\rangle + \sum_{m=1}^{M}\sum_i \alpha_{i,m} \\ &= \sum_m \left[-\frac{1}{2\lambda_m}\sum_i\sum_j \alpha_{i,m}\alpha_{j,m}K_{i,j} + \sum_i \alpha_{i,m} \right].\end{aligned}$$

Combining the non-negativity of $u_{i,m}$ and $v_{i,m}$ to (52) and (54), the nesting constraints (17) and the box constrains (16) can be obtained.

REFERENCES

[1] J. A. Hartigan, "Consistency of single linkage for high-density clusters," *Journal of the American Statistical 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 Transactions on Information Theory*, vol. 51, pp. 3806–3819, 2005.

[7] C. Scott and G. Blanchard, "Transductive anomaly detection," Tech. Rep., 2008, http://www.eecs.umich.edu/~cscott.

[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] 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.

[11] 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.

[12] R. H. S. H.-P. D. R. S. Agarwal, T. Graepel, "Generalization bounds for the area under the roc curve," *Journal of Machine Learning Research*, vol. 6, pp. 393–425, 2005.

[13] 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.

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

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

[16] 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.

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

[18] 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.

[19] 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.

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

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

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