THE ONE CLASS SUPPORT VECTOR MACHINE SOLUTION PATH

Gyemin Lee and Clayton D. Scott

Department of Electrical Engineering and Computer Science
University of Michigan, Ann Arbor, Michigan, USA
E-mail: {gyemin, cscott}@eecs.umich.edu

ABSTRACT

This paper applies the algorithm of Hastie et al. [1] to the problem of learning the entire solution path of the one class support vector machine (OC-SVM) as its free parameter ν varies from 0 to 1. The OC-SVM with Gaussian kernel is a nonparametric estimator of a level set of the density governing the observed sample, with the parameter ν implicitly defining the corresponding level. Thus, the path algorithm produces estimates of all level sets and can therefore be applied to a variety of problems requiring estimation of multiple level sets including clustering, outlier ranking, minimum volume set estimation, and density estimation. The algorithm's cost is comparable to the cost of computing the OC-SVM for a single point on the path. We introduce a heuristic for enforced nestedness of the sets in the path, and present a method for kernel bandwidth selection based in minimum integrated volume, a kind of AUC criterion. These methods are illustrated on three datasets.

Index Terms— support vector machines, one-class classification, solution path, density level set estimation

1. INTRODUCTION

The one class (or single class) support vector machine (OC-SVM) was introduced independently by Tax and Duin [2] and Schölkopf et al. [3] as an extension of the support vector classification methodology to the problem of one class classification. Recently Vert and Vert [4] proved that for the Gaussian kernel with bandwidth tending to zero, the OC-SVM is a consistent density level set estimator. The free parameter $\nu \in [0, 1]$ acts as an upper bound on the fraction of outlying points, and therefore affects which level set is estimated, although the mapping between ν and the corresponding density level is implicit and not known a priori.

In this paper we present an algorithm for learning the entire solution path of the OC-SVM as the parameter ν varies from 0 to 1. The algorithm relies on recasting the OC-SVM so that ν is replaced by another parameter C, analogous to the C in the original support vector classifier (SVC). This allows us to adapt a recent algorithm of Hastie et al. [1] for computing the solution path of the C-parametrized SVC. While C lacks an intuitive interpretation (unlike ν), this is irrelevant because the solution paths of the two formulations coincide. The solution path is piecewise linear in $1/C$ and can be computed efficiently. As C (or ν) varies from one extreme to another, so does the corresponding density level.

This work is motivated by a desire to perform nonparametric estimation of density level sets at a range of density levels. Since the path algorithm is about as costly as determining the OC-SVM at a fixed ν, the path algorithm offers considerable savings. The following applications are envisioned.

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" [5] that depicts the bifurcation of clusters into disjoint components and gives a hierarchical representation of cluster structure.

Outlier ranking: Given a dataset that may be contaminated with outliers/anomalies, a natural way to rank the data points in order of "outlyingness" (potential to be an outlier) is by the volume of the smallest density level set containing the point. Estimating the density level set at all levels allows one to prioritize the data points for further investigation of their status as outliers [6].

Minimum volume set estimation: A minimum volume set [7, 8] is a density level set that encloses a pre-specified probability mass of the distribution from which data are observed. Such sets are useful for outlier prediction with a guaranteed false alarm rate. Since neither ν nor C correspond to a precise mass enclosed, it is necessary to estimate level sets in a range and select the best by cross-validation or some other error estimate.

Density estimation: Estimating all the level sets of a density is equivalent to density estimation. In high dimensions, density estimates such as kernel density estimates are known to struggle while SVMs are touted for their ability to avoid overfitting. Our OC-SVM solution path may therefore offer advantages for density estimation in high dimensions, and such claims warrant further investigation.

2. ONE-CLASS SUPPORT VECTOR MACHINES

The OC-SVM was proposed in [2, 3] as a support vector methodology to estimate a set that encloses "most" of a given random sample $\{\mathbf{x}_i\}_{i=1}^n$, $\mathbf{x}_i \in \mathbb{R}^d$. Each $\mathbf{x}_i$ is first transformed via a map $\Phi : \mathbb{R}^d \to \mathcal{H}$ where $\mathcal{H}$ is a high (possibly infinite) dimensional Hilbert space generated by a positive-definite kernel $k(\mathbf{x}, \mathbf{x}')$. The kernel function corresponds to an inner product in $\mathcal{H}$ through $k(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle$. The OC-SVM attempts to find a hyperplane in the feature space that separates the data from the origin with maximum margin (the distance from the hyperplane to the origin). In the event that no such hyperplane exists, slack variables ξ_i allow for some points to be within the margin, and the free parameter $\nu \in [0, 1]$ controls the cost of such violations. In fact, ν can be shown to be an upper bound on the fraction of points within the margin (outliers) [3]. The hyperplane in feature space induces a generally nonlinear surface in the input space. In practice, the OC-SVM has only been successfully applied with the Gaussian kernel. For this kernel, the induced feature space is such that all points are mapped into the same orthant, and therefore the principle of separating the data from the origin is justified [9].

More precisely, the OC-SVM as presented in [3] solves the fol-

lowing quadratic program:

$$\min_{\mathbf{w},\boldsymbol{\xi},\rho} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{\nu n}\sum_{i=1}^{n}\xi_i - \rho \qquad (P_\nu)$$
$$\text{s.t.} \quad \langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle \geq \rho - \xi_i, \quad \xi_i \geq 0 \quad \text{for } i = 1, 2, \ldots, n.$$

The optimal $\mathbf{w} \in \mathcal{H}$ is the normal vector defining the hyperplane, and the function $g(\mathbf{x}) = \text{sgn}(\langle \mathbf{w}, \Phi(\mathbf{x})\rangle - \rho)$ determines whether a point is in $(+)$ or out $(-)$ of the estimated set. The quantity $\frac{\rho}{\|\mathbf{w}\|}$ is the margin, that is, the distance from the hyperplane $\{\mathbf{z} \in \mathcal{H} : \langle \mathbf{w}, \mathbf{z}\rangle = \rho\}$ to the origin.

In practice the quadratic program is solved via its dual, where we optimize over the Lagrange multipliers $\boldsymbol{\alpha}$:

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j k(\mathbf{x}_i, \mathbf{x}_j) \qquad (D_\nu)$$
$$\text{s.t.} \quad 0 \leq \alpha_i \leq \frac{1}{\nu n}, \quad \sum_i \alpha_i = 1.$$

The optimal normal vector is given by $\mathbf{w} = \sum_i \alpha_i \Phi(\mathbf{x}_i)$.

The points $\mathbf{x}_i$ for which $\alpha_i \neq 0$ are called support vectors. It can also be shown that ν lower bounds the fraction of support vectors.

The path algorithm is facilitated by an alternative formulation of the OC-SVM which replaces ν with a different parameter $C \geq 0$:

$$\min_{\mathbf{w},\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \qquad (P_C)$$
$$\text{s.t.} \quad \langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad \text{for } i = 1, 2, \ldots, n$$

with its dual

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j k(\mathbf{x}_i, \mathbf{x}_j) - \sum_i \alpha_i \qquad (D_C)$$
$$\text{s.t.} \quad 0 \leq \alpha_i \leq C.$$

The two formulations P_ν and P_C are equivalent in the sense of the following result. The proof follows a similar course to [9].

Proposition 1. *If P_ν results in $\rho > 0$, then P_C with $C = \frac{1}{\nu n \rho}$ leads to the same decision function.*

Proof. Suppose that $\mathbf{w}_0, \boldsymbol{\xi}_0$, and ρ_0 solve P_ν. Then the $\mathbf{w}_0, \boldsymbol{\xi}_0$ also minimize the objective function of P_C, with $C = 1/\nu n$, subject to the constraints of P_ν with $\rho = \rho_0$. Letting $\mathbf{w} = \rho_0\mathbf{w}', \boldsymbol{\xi} = \rho_0\boldsymbol{\xi}'$ in P_ν, we see that $\mathbf{w}_0, \boldsymbol{\xi}_0$ optimize the objective function (scaled by ρ_0^2) of P_C with $C = \frac{1}{\nu n \rho_0}$, subject to the constraints of P_C. □

Although C lacks the interpretation of ν as a bound on the fraction of outliers, the solution paths of the two quadratic programs are the same. While the C parametrization facilitates the path algorithm, it should be possible to reparametrize the path, once learned, in terms of ν using connections established in [10].

3. PATH ALGORITHM

Hastie et al. [1] demonstrated that the Lagrange multipliers of the SVC are piecewise-linear in $1/C$, and developed an algorithm for finding this solution path. The computational complexity of the algorithm is on the order of the complexity of finding a single point on the path. We adapt their approach, using the same notation, to develop a similar path algorithm for the OC-SVM. Indeed, the OC-SVM may be viewed as the application of the SVC to an augmented dataset, where the original $\mathbf{x}_i$ constitute one class and their reflections about the origin $-\mathbf{x}_i$ constitute the other. Because of the structure of this reduction, however, the path algorithm simplifies somewhat. For example, path initialization for the OC-SVM can be significantly easier that for the SVC.

Introducing the parameter $\lambda = \frac{1}{C}$, we can rewrite P_C as

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

with the solution

$$\mathbf{w} = \frac{1}{\lambda}\sum_i \alpha_i \Phi(\mathbf{x}_i) \qquad (1)$$

corresponding to a decision function $g(\mathbf{x}) = \text{sgn}(\langle \mathbf{w}, \Phi(\mathbf{x})\rangle - 1)$.

From the Karush-Khun-Tucker conditions (here β_i is the Lagrange multiplier corresponding to the constraint $\xi_i \geq 0$)

$$\alpha_i(f(\mathbf{x}_i) - 1 + \xi_i) = 0, \qquad \beta_i\xi_i = 0,$$

it follows that

$$\begin{array}{llll} f(\mathbf{x}_i) > 1 & \Rightarrow & \xi_i = 0, & \alpha_i = 0 \\ f(\mathbf{x}_i) = 1 & \Rightarrow & \xi_i = 0, & \alpha_i \in [0, 1] \\ f(\mathbf{x}_i) < 1 & \Rightarrow & \xi_i > 0, & \alpha_i = 1 \end{array}$$

where $f(\mathbf{x}) = \langle \mathbf{w}, \Phi(\mathbf{x})\rangle$. Then $\{\Phi(\mathbf{x}) : f(\mathbf{x}) = 1\}$ defines a hyperplane with a distance $\frac{1}{\|\mathbf{w}\|}$ from the origin.

Decreasing λ from a large value toward zero yields the entire solution path. As λ decreases, $\|\mathbf{w}\|$ increases, and hence the margin width decreases. As this width decreases, points cross the margin $(f(\mathbf{x}_i) = 1)$ and move from inside $(f(\mathbf{x}_i) < 1)$ to outside the margin $(f(\mathbf{x}_i) > 1)$ while their corresponding α_i change from 1 to 0. During this process, the algorithm monitors the following subsets:

- $\mathcal{R} = \{i : f(\mathbf{x}_i) > 1,\ \alpha_i = 0\}$.
- $\mathcal{E} = \{i : f(\mathbf{x}_i) = 1,\ 0 \leq \alpha_i \leq 1\}$,
- $\mathcal{L} = \{i : f(\mathbf{x}_i) < 1,\ \alpha_i = 1\}$.

3.1. Initialization

Since the dataset belongs to a single class, the initialization of the OC-SVM is easier than the two-class SVM, where the process depends on whether the classes are balanced or not [1].

For sufficiently large λ, $\mathbf{w}$ vanishes from (1). Then the margin width tends to infinity and all the data falls inside the margin; thus, $f(\mathbf{x}_i) \leq 1$ and $\alpha_i = 1$ for $\forall i$. For large values of λ, we have

$$f(\mathbf{x}_i) = \frac{1}{\lambda}\langle \mathbf{w}^*, \Phi(\mathbf{x}_i)\rangle \leq 1, \qquad \forall i$$

where $\mathbf{w}^* = \sum_i \Phi(\mathbf{x}_i)$. Finding the most extreme point from the origin, we can obtain the initial value of λ, $\lambda_0 = \langle \mathbf{w}^*, \Phi(\mathbf{x}_{i_+})\rangle$, where $i_+ = \arg\max_i \langle \mathbf{w}^*, \Phi(\mathbf{x}_i)\rangle$.

3.2. Tracing the path

As λ decreases, the algorithm keeps track of the following events:

A. A point enters $\mathcal{E}$ from $\mathcal{L}$ or $\mathcal{R}$.

B. A point leaves $\mathcal{E}$ and joins either $\mathcal{R}$ or $\mathcal{L}$.

We let α_j^l and λ_l denote the parameters right after the lth event and $f^l(\mathbf{x})$ the function at this point. Define $\mathcal{E}_l$ similarly and suppose $|\mathcal{E}_l| = m$. Since

$$f(\mathbf{x}) = \frac{1}{\lambda}\sum_{j=1}^{n}\alpha_j k(\mathbf{x}_j, \mathbf{x}),$$

for $\lambda_l > \lambda > \lambda_{l+1}$ we have

$$\begin{aligned} f(\mathbf{x}) &= \left[f(\mathbf{x}) - \frac{\lambda_l}{\lambda}f^l(\mathbf{x})\right] + \frac{\lambda_l}{\lambda}f^l(\mathbf{x}) \\ &= \frac{1}{\lambda}\left[\sum_{j\in\mathcal{E}_l}(\alpha_j - \alpha_j^l)k(\mathbf{x}, \mathbf{x}_j) + \lambda_l f^l(\mathbf{x})\right]. \end{aligned} \quad (2)$$

The last equality holds because for this range of λ only points in $\mathcal{E}_l$ change their α_j, while all other points in $\mathcal{R}_l$ or $\mathcal{L}_l$ have fixed $\alpha_j = 0$ or 1, respectively. Since $f(\mathbf{x}_i) = 1$ for all $i \in \mathcal{E}_l$, we have

$$\sum_{j\in\mathcal{E}_l}\delta_j k(\mathbf{x}_i, \mathbf{x}_j) = \lambda_l - \lambda, \qquad \forall i \in \mathcal{E}_l$$

where $\delta_j = \alpha_j^l - \alpha_j$.

Now let $\mathbf{K}_l$ be the $m \times m$ matrix such that $[\mathbf{K}_l]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ for $i, j \in \mathcal{E}_l$. Then we have $\mathbf{K}_l\boldsymbol{\delta} = (\lambda_l - \lambda)\mathbf{1}$. If $\mathbf{K}_l$ has full rank, we obtain $\mathbf{b} = \mathbf{K}_l^{-1}\mathbf{1}$, and hence

$$\alpha_j = \alpha_j^l - (\lambda_l - \lambda)b_j, \qquad j \in \mathcal{E}_l. \quad (3)$$

Substituting this result into (2), we have

$$f(\mathbf{x}) = \frac{\lambda_l}{\lambda}[f^l(\mathbf{x}) - h^l(\mathbf{x})] + h^l(\mathbf{x}) \quad (4)$$

where $h^l(\mathbf{x}) = \sum_{j\in\mathcal{E}_l} b_j k(\mathbf{x}, \mathbf{x}_j)$. Therefore, the α_j for $j \in \mathcal{E}$ are piecewise-linear in λ. If $\mathbf{K}_l$ is not invertible, some of the α_i have non-unique paths. These cases are rare in practice and discussed more in [1].

3.3. Finding the next breakpoint

The $(l+1)$-st event occurs when:

A. Some $\mathbf{x}_j$ for which $j \in \mathcal{L}_l \cup \mathcal{R}_l$ hits the hyperplane, meaning $f(\mathbf{x}_j) = 1$. Then, from (4), we know that

$$\lambda = \lambda_l\frac{f^l(\mathbf{x}_j) - h^l(\mathbf{x}_j)}{1 - h^l(\mathbf{x}_j)}.$$

B. Some α_j for which $j \in \mathcal{E}_l$ reaches 0 or 1. In this case, from (3), we know, respectively, that

$$\lambda = \frac{-\alpha_j^l + \lambda_l b_j}{b_j}, \quad \lambda = \frac{1 - \alpha_j^l + \lambda_l b_j}{b_j}.$$

The next event corresponds to the largest λ such that $\lambda < \lambda_l$.

3.4. Obtaining nested density level set estimates

As discussed above, we interpret a decision function through the path algorithm as a density level set estimator:

$$\widehat{G}_\lambda = \{\mathbf{x} : f_\lambda(\mathbf{x}) = \frac{1}{\lambda}\sum_i \alpha_i(\lambda)k(\mathbf{x}_i, \mathbf{x}) > 1\}.$$

Note that λ does not correspond to the density level, but rather to the parameter in the OC-SVM.

Since density level sets are nested, it seems reasonable to enforce level set estimators to be nested as well. The experiments in the next section, however, show that this condition does not hold in general for the OC-SVM. To impose nestedness, we modify the output of the path algorithm by introducing

$$\widehat{G}'_\lambda = \bigcup_{\mu\geq\lambda}\widehat{G}_\mu,$$

which ensures nestedness of the path of sets. Once $\mathbf{x}$ falls into a density level set ($f_\lambda(\mathbf{x}) > 1$), therefore, it remains in the set.

3.5. Bandwidth selection via minimum integrated volume

Correctly setting the bandwidth of the Gaussian kernel in the OC-SVM is critical to its performance. We propose to select the bandwidth minimizing the integrated volume (IV), defined as follows. The family of estimates $\widehat{G}_\lambda$ yields a curve $(P(\widehat{G}_\lambda), \mu(\widehat{G}_\lambda))$ as λ varies, where P the underlying probability measure and μ denotes Euclidean volume. The IV is the area under this curve. Since level sets of a density are minimum volume sets [7], meaning they have the smallest volume for the mass they enclose, the true density has the smallest possible integrated volume. Selecting the bandwidth by minimum integrated volume (MIV) thus attempts to do a good job of approximating the true level sets across the whole range of different levels/enclosed masses. In our implementation we estimate mass via cross-validation and volume by a simulated uniform sample on a box enclosing the data. We also note that MIV is equivalent to maximizing the area under the ROC curve (AUC) corresponding to the null distribution P and a uniform alternative.

4. EXPERIMENTS

To implement the OC-SVM solution path algorithm, we adapted the SvmPath package [11]. We examined three random data sets "mixture", "multi" and "ring" each with 200 data points. The first data set is from [1], "multi" is a three component Gaussian mixture distribution with unequal weights, and "ring" is a ring-shaped dataset $\{(r_i, \theta_i)\}_{i=1}^{200}$ such that the radius r_i is drawn from a Rayleigh distribution with an offset and the angle θ_i is drawn from unform distribution.

In our experiments, the radial basis function (Gaussian) kernel $k(\mathbf{x}, \mathbf{x}') = \exp(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2})$ was used. Fig. 1 illustrates four mass-volume curves of "multi". To estimate mass and volume of a density level set, we used 5-fold cross validation. Among these curves, $\sigma = 1$ achieves the minimum integrated volume. Following the discussion above, we searched for bandwidth σ over the logarithmically spaced grid of 30 points from 0.3 to 3. Integrated volumes for each value of σ can be seen in Fig. 2. We can observe well-defined minimum integrated volume near $\sigma = 1$.

The results using σ with minimum integrated volumes are presented in Fig. 3. In the figure, small circles represent data points and solid lines depict the boundary of the decision function. The region inside the boundary corresponds to a density level set estimate.

The images in the left column show $\widehat{G}_\lambda$ for the final value of λ, for each dataset. We see that $\widehat{G}_\lambda$ has holes where clearly it should not. In the right column, each image contains five different nested set estimates $\widehat{G}'_\lambda$ for various increments of λ along the path. The holes no longer appear.

Movies illustrating the path algorithm with different kernel widths and for the three datasets are available in `http:`

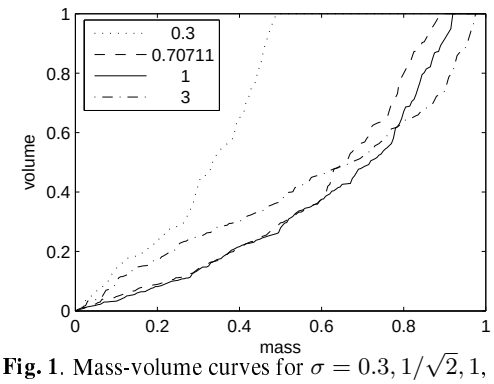


Fig. 1. Mass-volume curves for $\sigma = 0.3, 1/\sqrt{2}, 1$, and 3

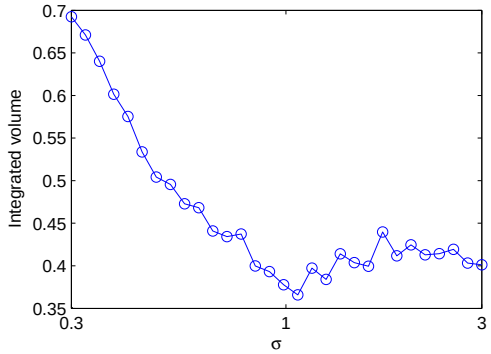


Fig. 2. Integrated volume of "multi" dataset with respect to kernel bandwidth σ

`//www-personal.umich.edu/~gyemin/ocsvm/`. These movies clearly illustrate the non-nested nature of the OC-SVM path $\widehat{G}_\lambda$ and the monotone growth of the nested sets $\widehat{G}'_\lambda$. They also clearly demonstrate the effect of different kernel widths: small widths lead to overfitting (many holes in the estimated sets) while large widths lead to overly rigid shapes that fail to capture the contours of the density.

5. CONCLUSION

In this paper, we have applied the work of Hastie et al. [1] to compute the entire solution path for the OC-SVM. The key step allowing our adaptation was a reformulation of the OC-SVM in terms of parameter λ in which the path is piecewise linear. The path algorithm yields a family of density level set estimators. We demonstrated a simple heuristic for enforcing nestedness, and developed a minimum integrated volume criterion for kernel bandwidth selection.

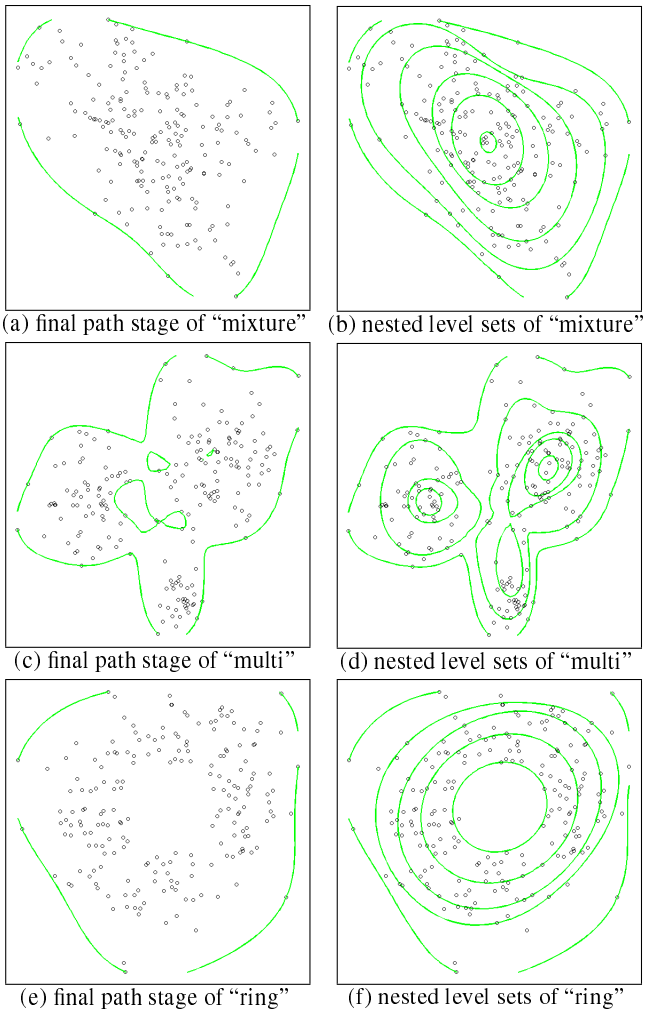
Future work may include (1) applying our methodology to the problems outlined in the introduction; (2) comparing to other multiple level set estimators such as kernel density estimation followed by thresholding; (3) other methods for enforcing nestedness, such as incorporating nestedness into the solution path algorithm.

6. REFERENCES

[1] 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.

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

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



(a) final path stage of "mixture" (b) nested level sets of "mixture"

(c) final path stage of "multi" (d) nested level sets of "multi"

(e) final path stage of "ring" (f) nested level sets of "ring"

Fig. 3. Examples of OC-SVM for three datasets. The rows correspond to the "mixture", "multi" and "ring" datasets respectively. (a), (c) and (e) show the final stage of the path algorithm. (b), (d) and (f) show several stages of the path after the nesting construction in Section 3.4 has been enforced.

[4] 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.

[5] 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.

[6] C. Scott and E. Kolacyzk, "Simulataneous nonparametric annotation of contaminated multivariate data," 2007, in preparation.

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

[8] M. Davenport, R. Baraniuk, and C. Scott, "Learning minimum volume sets with support vector machines," in *IEEE Int. Workshop on Machine Learning for Signal Processing*, Maynooth, Ireland, Sept 2006.

[9] B. Schölkopf and A.J. Smola, *Learning with Kernels*, chapter 7, pp. 208–209, MIT Press, Cambridge, MA, 2002.

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

[11] T. Hastie, "SvmPath: fit the entire regularization path for the SVM," http://www-stat.stanford.edu/ hastie/Papers/SVMPATH/, 2004.