Estimating the Support of a High-Dimensional Distribution

Bernhard Schölkopf[⋆], John C. Platt[‡],
John Shawe-Taylor[†], Alex J. Smola[§],
Robert C. Williamson[§],

[⋆] Microsoft Research Ltd, 1 Guildhall Street, Cambridge CB2 3NH, UK
[‡] Microsoft Research, 1 Microsoft Way, Redmond, WA, USA
[†] Royal Holloway, University of London, Egham, UK
[§] Department of Engineering, Australian National University,
Canberra 0200, Australia

bsc@scientist.com, jplatt@microsoft.com, john@dcs.rhbnc.ac.uk
Alex.Smola@anu.edu.au, Bob.Williamson@anu.edu.au

27 November 1999; revised 21 August 2000

Technical Report
MSR-TR-99-87

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

An abridged version of this document will appear in *Neural Computation*.

Abstract

Suppose you are given some dataset drawn from an underlying probability distribution P and you want to estimate a "simple" subset S of input space such that the probability that a test point drawn from P lies outside of S equals some a priori specified value between 0 and 1.

We propose a method to approach this problem by trying to estimate a function f which is positive on S and negative on the complement. The functional form of f is given by a kernel expansion in terms of a potentially small subset of the training data; it is regularized by controlling the length of the weight vector in an associated feature space. The expansion coefficients are found by solving a quadratic programming problem, which we do by carrying out sequential optimization over pairs of input patterns. We also provide a theoretical analysis of the statistical performance of our algorithm.

The algorithm is a natural extension of the support vector algorithm to the case of unlabelled data.

Keywords. Support Vector Machines, Kernel Methods, Density Estimation, Unsupervised Learning, Novelty Detection, Condition Monitoring, Outlier Detection

1 Introduction

During recent years, a new set of kernel techniques for supervised learning has been developed (Vapnik, 1995; Schölkopf et al., 1999a). Specifically, support vector (SV) algorithms for pattern recognition, regression estimation and solution of inverse problems have received considerable attention.

There have been a few attempts to transfer the idea of using kernels to compute inner products in feature spaces to the domain of unsupervised learning. The problems in that domain are, however, less precisely specified. Generally, they can be characterized as estimating *functions* of the data which tell you something interesting about the underlying distributions. For instance, kernel PCA can be characterized as computing functions which on the training data produce unit variance outputs while having minimum norm in feature space (Schölkopf et al., 1999b). Another kernel-based unsupervised learning technique, regularized principal manifolds (Smola et al., 2000), computes functions which give a mapping onto a lower-dimensional manifold minimizing a regularized quantization error. Clustering algorithms are further examples of unsupervised learning techniques which can be kernelized (Schölkopf et al., 1999b).

An extreme point of view is that unsupervised learning is about estimating densities. Clearly, knowledge of the density of P would then allow us to solve whatever problem can be solved on the basis of the data.

The present work addresses an easier problem: it proposes an algorithm which computes a binary function which is supposed to capture regions in input space where the probability density lives (its support), i.e. a function such that most of the data will live in the region where the function is nonzero (Schölkopf et al., 1999). In doing so, it is in line with Vapnik's principle never to solve a problem which is more general than the one we actually need to solve. Moreover, it is applicable also in cases where the density of the data's distribution is not even well-defined, e.g. if there are singular components.

The article is organized as follows. After a review of some previous work in Sec. 2, we propose SV algorithms for the considered problem. Sec. 4 gives details on the implementation of the optimization procedure, followed by theoretical results characterizing the present approach. In Sec. 6, we apply the algorithm to artificial as well as real-world data. We conclude with a discussion.

2 Previous Work

In order to describe some previous work, it is convenient to introduce the following definition of a (multi-dimensional) quantile function, introduced by Einmal and Mason (1992). Let $\mathbf{x}_1, \ldots, \mathbf{x}_\ell$ be i.i.d. random variables in a set $\mathcal{X}$ with distribution P. Let $\mathcal{C}$ be a class of measurable subsets of $\mathcal{X}$ and let λ be a real-valued function defined on $\mathcal{C}$. The *quantile function* with respect to $(P, \lambda, \mathcal{C})$ is

$$U(\alpha) = \inf\{\lambda(C) \colon P(C) \geq \alpha, C \in \mathcal{C}\} \quad 0 < \alpha \leq 1.$$

In the special case where P is the empirical distribution ($P_\ell(C) := \frac{1}{\ell} \sum_{i=1}^{\ell} \mathbf{1}_C(\mathbf{x}_i)$), we obtain the *empirical quantile function*. We denote by $C(\alpha)$ and $C_\ell(\alpha)$ the (not necessarily unique) $C \in \mathcal{C}$ that attains the infimum (when it is achievable). The most common choice of λ is Lebesgue measure, in which case $C(\alpha)$ is the minimum volume $C \in \mathcal{C}$ that contains at least a fraction α of the probability mass. Estimators of the form $C_\ell(\alpha)$ are called *minimum volume estimators*.

Observe that for $\mathcal{C}$ being all Borel measurable sets, $C(1)$ is the *support* of the density p corresponding to P, assuming it exists. (Note that $C(1)$ is well defined even when p does not exist.) For smaller classes $\mathcal{C}$, $C(1)$ is the minimum volume $C \in \mathcal{C}$ containing the support of p.

Turning to the case where $\alpha < 1$, it seems the first work was reported by Sager (1979) and then Hartigan (1987) who considered $\mathcal{X} = \mathbb{R}^2$ with $\mathcal{C}$ being the class of closed convex sets in $\mathcal{X}$. (They actually considered density contour clusters; cf. Appendix A for a definition.) Nolan (1991) considered higher dimensions with $\mathcal{C}$ being the class of ellipsoids. Tsybakov (1997) has studied an estimator based on piecewise polynomial approximation of $C(\alpha)$ and has shown it attains the asymptotically minimax rate for certain classes of densities p. Polonik (1997) has studied the estimation of $C(\alpha)$ by $C_\ell(\alpha)$. He derived asymptotic rates of convergence in terms of various measures of richness of $\mathcal{C}$. He considered both VC classes and classes with a log ϵ-covering number with bracketing of order $O(\epsilon^{-r})$ for $r > 0$. More information on minimum volume estimators can be found in that work, and in Appendix A.

A number of applications have been suggested for these techniques. They include problems in medical diagnosis (Tarassenko et al., 1995), marketing (Ben-David and Lindenbaum, 1997), condition monitoring of machines (Devroye and Wise, 1980), estimating manufacturing yields (Stoneking, 1999), econometrics and generalized nonlinear principal curves (Tsybakov, 1997; Korostelev and Tsybakov, 1993), regression and spectral analysis (Polonik, 1997), tests for multimodality and clustering (Polonik, 1995b) and others (Müller, 1992).

Most of these papers, in particular those with a theoretical slant, do not go all the way in devising practical algorithms that work on high-dimensional real-world-problems. A notable exception to this is the work of Tarassenko et al. (1995).

Polonik (1995a) has shown how one can use estimators of $C(\alpha)$ to construct density estimators. The point of doing this is that it allows one to encode a range of prior assumptions about the true density p that would be impossible to do within the traditional density estimation framework. He has shown asymptotic consistency and rates of convergence for densities belonging to VC-classes or with a known rate of growth of metric entropy with bracketing.

Let us conclude this section with a short discussion of how the present work relates to the above. The present paper describes an algorithm which finds regions close to $C(\alpha)$. Our class $\mathcal{C}$ is defined implicitly via a kernel k as the set of half-spaces in a SV feature space. We do not try to minimize the volume of C in input space. Instead, we minimize a SV style regularizer which, using a kernel, controls the smoothness of the estimated function describing C. In terms of multi-dimensional quantiles, our approach can be thought of as employing $\lambda(C_w) = \|w\|^2$, where $C_w = \{x\colon f_w(x) \geq \rho\}$. Here, (w, ρ) are a weight vector and an offset parametrizing a hyperplane in the feature space associated with the kernel.

The main contribution of the present work is that we propose an algorithm that has tractable computational complexity, even in high-dimensional cases. Our theory, which uses very similar tools to those used by Polonik, gives results that we expect will be of more use in a finite sample size setting.

3 Algorithms

We first introduce terminology and notation conventions. We consider training data

$$\mathbf{x}_1, \ldots, \mathbf{x}_\ell \in \mathcal{X}, \tag{1}$$

where $\ell \in \mathbb{N}$ is the number of observations, and $\mathcal{X}$ is some set. For simplicity, we think of it as a compact subset of $\mathbb{R}^N$. Let Φ be a feature map $\mathcal{X} \to F$, i.e. a map into an inner product space F such that the inner product in the image of Φ can be computed by evaluating some simple kernel (Boser et al., 1992; Vapnik, 1995; Schölkopf et al., 1999a)

$$k(\mathbf{x}, \mathbf{y}) = (\Phi(\mathbf{x}) \cdot \Phi(\mathbf{y})), \tag{2}$$

such as the Gaussian kernel

$$k(\mathbf{x}, \mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2/c}. \tag{3}$$

Indices i and j are understood to range over $1, \ldots, \ell$ (in compact notation: $i, j \in [\ell]$). Bold face greek letters denote ℓ-dimensional vectors whose components are labelled using normal face type.

In the remainder of this section, we shall develop an algorithm which returns a function f that takes the value $+1$ in a "small" region capturing most of the data points, and -1 elsewhere. Our strategy is to map the data into the feature space corresponding to the kernel, and to separate them from the origin with maximum margin. For a new point $\mathbf{x}$, the value $f(\mathbf{x})$ is determined by evaluating which side of the hyperplane it falls on, in feature space. Via the freedom to utilize different types of kernel functions,

this simple geometric picture corresponds to a variety of nonlinear estimators in input space.

To separate the data set from the origin, we solve the following quadratic program:

$$\min_{w \in F, \boldsymbol{\xi} \in \mathbb{R}^\ell, \rho \in \mathbb{R}} \quad \tfrac{1}{2}\|w\|^2 + \tfrac{1}{\nu\ell}\textstyle\sum_i \xi_i - \rho \tag{4}$$

$$\text{subject to} \quad (w \cdot \Phi(\mathbf{x}_i)) \geq \rho - \xi_i, \quad \xi_i \geq 0. \tag{5}$$

Here, $\nu \in (0, 1]$ is a parameter whose meaning will become clear later.

Since nonzero slack variables ξ_i are penalized in the objective function, we can expect that if w and ρ solve this problem, then the decision function

$$f(\mathbf{x}) = \operatorname{sgn}((w \cdot \Phi(\mathbf{x})) - \rho) \tag{6}$$

will be positive for most examples $\mathbf{x}_i$ contained in the training set,[1] while the SV type regularization term $\|w\|$ will still be small. The actual trade-off between these two goals is controlled by ν.

Using multipliers $\alpha_i, \beta_i \geq 0$, we introduce a Lagrangian

$$L(w, \boldsymbol{\xi}, \rho, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\|w\|^2 + \frac{1}{\nu\ell}\sum_i \xi_i - \rho - \sum_i \alpha_i((w \cdot \Phi(\mathbf{x}_i)) - \rho + \xi_i) - \sum_i \beta_i \xi_i, \tag{7}$$

and set the derivatives with respect to the primal variables $w, \boldsymbol{\xi}, \rho$ equal to zero, yielding

$$w = \sum_i \alpha_i \Phi(\mathbf{x}_i), \tag{8}$$

$$\alpha_i = \frac{1}{\nu\ell} - \beta_i \leq \frac{1}{\nu\ell}, \quad \sum_i \alpha_i = 1. \tag{9}$$

In (8), all patterns $\{\mathbf{x}_i : i \in [\ell],\ \alpha_i > 0\}$ are called Support Vectors. Together with (2), the SV expansion transforms the decision function (6) into a kernel expansion

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_i \alpha_i k(\mathbf{x}_i, \mathbf{x}) - \rho\right). \tag{10}$$

Substituting (8) – (9) into L (7), and using (2), we obtain the dual problem:

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\sum_{ij} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) \quad \text{subject to } \ 0 \leq \alpha_i \leq \frac{1}{\nu\ell}, \quad \sum_i \alpha_i = 1. \tag{11}$$

One can show that at the optimum, the two inequality constraints (5) become equalities if α_i and β_i are nonzero, i.e. if $0 < \alpha_i < 1/(\nu\ell)$. Therefore, we can recover ρ by exploiting that for any such α_i, the corresponding pattern $\mathbf{x}_i$ satisfies

$$\rho = (w \cdot \Phi(\mathbf{x}_i)) = \sum_j \alpha_j k(\mathbf{x}_j, \mathbf{x}_i). \tag{12}$$

[1] We use the convention that $\operatorname{sgn}(z)$ equals 1 for $z \geq 0$ and -1 otherwise.

Note that if ν approaches 0, the upper boundaries on the Lagrange multipliers tend to infinity, i.e. the second inequality constraint in (11) becomes void. The problem then resembles the corresponding *hard margin* algorithm, since the penalization of errors becomes infinite, as can be seen from the primal objective function (4). It is still a feasible problem, since we have placed no restriction on the offset ρ, so it can become a large negative number in order to satisfy (5). If we had required $\rho \geq 0$ from the start, we would have ended up with the constraint $\sum_i \alpha_i \geq 1$ instead of the corresponding equality constraint in (11), and the multipliers α_i could have diverged.

It is instructive to compare (11) to a Parzen windows estimator. To this end, suppose we use a kernel which can be normalized as a density in input space, such as the Gaussian (3). If we use $\nu = 1$, then the two constraints only allow the solution $\alpha_1 = \ldots = \alpha_\ell = 1/\ell$. Thus the kernel expansion in (10) reduces to a Parzen windows estimate of the underlying density. For $\nu < 1$, the equality constraint in (11) still ensures that the decision function is a thresholded density, however, in that case, the density will only be represented by a subset of training examples (the SVs) — those which are important for the decision (10) to be taken. Sec. 5 will explain the precise meaning of ν.

To conclude this section, we note that one can also use *balls* to describe the data in feature space, close in spirit to the algorithms of Schölkopf et al. (1995), with hard boundaries, and Tax and Duin (1999), with "soft margins." Again, we try to put *most of* the data into a small ball by solving, for $\nu \in (0, 1)$,

$$\begin{aligned} \min_{R \in \mathbb{R}, \boldsymbol{\xi} \in \mathbb{R}^\ell, c \in F} \quad & R^2 + \tfrac{1}{\nu\ell} \textstyle\sum_i \xi_i \\ \text{subject to} \quad & \|\Phi(\mathbf{x}_i) - c\|^2 \leq R^2 + \xi_i, \quad \xi_i \geq 0 \quad \text{for } i \in [\ell]. \end{aligned} \tag{13}$$

This leads to the dual

$$\min_{\boldsymbol{\alpha}} \quad \textstyle\sum_{ij} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) - \sum_i \alpha_i k(\mathbf{x}_i, \mathbf{x}_i) \tag{14}$$

$$\text{subject to} \quad 0 \leq \alpha_i \leq \tfrac{1}{\nu\ell}, \quad \textstyle\sum_i \alpha_i = 1 \tag{15}$$

and the solution

$$c = \sum_i \alpha_i \Phi(\mathbf{x}_i), \tag{16}$$

corresponding to a decision function of the form

$$f(\mathbf{x}) = \operatorname{sgn}\left(R^2 - \sum_{ij} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) + 2 \sum_i \alpha_i k(\mathbf{x}_i, \mathbf{x}) - k(\mathbf{x}, \mathbf{x}) \right). \tag{17}$$

Similar to the above, R^2 is computed such that for any $\mathbf{x}_i$ with $0 < \alpha_i < 1/(\nu\ell)$ the argument of the sgn is zero.

For kernels $k(\mathbf{x}, \mathbf{y})$ which only depend on $\mathbf{x} - \mathbf{y}$, $k(\mathbf{x}, \mathbf{x})$ is constant. In this case, the equality constraint implies that the linear term in the dual target function is constant, and the problem (14–15) turns out to be equivalent to (11). It can be shown that the same holds true for the decision function, hence the two algorithms coincide in that

case. This is geometrically plausible: for constant $k(\mathbf{x}, \mathbf{x})$, all mapped patterns lie on a sphere in feature space. Therefore, finding the smallest sphere (containing the points) really amounts to finding the smallest segment of the sphere that the data live on. The segment, however, can be found in a straightforward way by simply intersecting the data sphere with a hyperplane — the hyperplane with maximum margin of separation to the origin will cut off the smallest segment.

4 Optimization

The last section has formulated quadratic programs (QPs) for computing regions that capture a certain fraction of the data. These constrained optimization problems can be solved via an off-the-shelf QP package to compute the solution. They do, however, possess features that set them apart from generic QPs, most notably the simplicity of the constraints. In the present section, we describe an algorithm which takes advantage of these features and empirically scales better to large data set sizes than a standard QP solver with time complexity of order $O(\ell^3)$ (cf. Platt, 1999). The algorithm is a modified version of SMO (Sequential Minimal Optimization), an SV training algorithm originally proposed for classification (Platt, 1999), and subsequently adapted to regression estimation (Smola and Schölkopf, 2000).

The strategy of SMO is to break up the constrained minimization of (11) into the smallest optimization steps possible. Due to the constraint on the sum of the dual variables, it is impossible to modify individual variables separately without possibly violating the constraint. We therefore resort to optimizing over pairs of variables.

Elementary optimization step. For instance, consider optimizing over α_1 and α_2 with all other variables fixed. Using the shorthand $K_{ij} := k(\mathbf{x}_i, \mathbf{x}_j)$, (11) then reduces to

$$\min_{\alpha_1, \alpha_2} \frac{1}{2} \sum_{i,j=1}^{2} \alpha_i \alpha_j K_{ij} + \sum_{i=1}^{2} \alpha_i C_i + C, \tag{18}$$

with $C_i := \sum_{j=3}^{\ell} \alpha_j K_{ij}$ and $C := \sum_{i,j=3}^{\ell} \alpha_i \alpha_j K_{ij}$, subject to

$$0 \leq \alpha_1, \alpha_2 \leq \frac{1}{\nu \ell}, \quad \sum_{i=1}^{2} \alpha_i = \Delta, \tag{19}$$

where $\Delta := 1 - \sum_{i=3}^{\ell} \alpha_i$.

We discard C, which is independent of α_1 and α_2, and eliminate α_1 to obtain

$$\min_{\alpha_2} \frac{1}{2}(\Delta - \alpha_2)^2 K_{11} + (\Delta - \alpha_2)\alpha_2 K_{12} + \frac{1}{2}\alpha_2^2 K_{22} + (\Delta - \alpha_2) C_1 + \alpha_2 C_2, \tag{20}$$

with the derivative

$$-(\Delta - \alpha_2) K_{11} + (\Delta - 2\alpha_2) K_{12} + \alpha_2 K_{22} - C_1 + C_2. \tag{21}$$

Setting this to zero and solving for α_2, we get

$$\alpha_2 = \frac{\Delta (K_{11} - K_{12}) + C_1 - C_2}{K_{11} + K_{22} - 2K_{12}}. \tag{22}$$

Once α_2 is found, α_1 can be recovered from $\alpha_1 = \Delta - \alpha_2$. If the new point (α_1, α_2) is outside of $[0, 1/(\nu\ell)]$, the constrained optimum is found by projecting α_2 from (22) into he region allowed by the constraints, and the re-computing α_1.

The offset ρ is recomputed after every such step.

Additional insight can be obtained by rewriting the last equation in terms of the outputs of the kernel expansion on the examples $\mathbf{x}_1$ and $\mathbf{x}_2$ before the optimization step. Let α_1^*, α_2^* denote the values of their Lagrange parameter before the step. Then the corresponding outputs (cf. (10)) read

$$O_i := K_{1i}\alpha_1^* + K_{2i}\alpha_2^* + C_i. \tag{23}$$

Using the latter to eliminate the C_i, we end up with an update equation for α_2 which does not explicitly depend on α_1^*,

$$\alpha_2 = \alpha_2^* + \frac{O_1 - O_2}{K_{11} + K_{22} - 2K_{12}}, \tag{24}$$

which shows that the update is essentially the fraction of first and second derivative of the objective function along the direction of ν-constraint satisfaction.

Clearly, the same elementary optimization step can be applied to any pair of two variables, not just α_1 and α_2. We next briefly describe how to do the overall optimization.

Initialization of the algorithm. We start by setting a fraction ν of all α_i, randomly chosen, to $1/(\nu\ell)$. If $\nu\ell$ is not an integer, then one of the examples is set to a value in $(0, 1/(\nu\ell))$ to ensure that $\sum_i \alpha_i = 1$. Moreover, we set the initial ρ to $\max\{O_i : i \in [\ell],\ \alpha_i > 0\}$.

Optimization algorithm. We then select a first variable for the elementary optimization step in one of the two following ways. Here, we use the shorthand SV_{nb} for the indices of variables which are not at bound, i.e. $SV_{nb} := \{i : i \in [\ell],\ 0 < \alpha_i < 1/(\nu\ell)\}$. At the end, these correspond to points that will sit exactly on the hyperplane, and that will therefore have a strong influence on its precise position.

(i) We scan over the entire data set[2] until we find a variable violating a KKT condition (Bertsekas, 1995, e.g.), i.e. a point such that $(O_i - \rho) \cdot \alpha_i > 0$ or $(\rho - O_i) \cdot (1/(\nu\ell) - \alpha_i) > 0$. Once we have found one, say α_i, we pick α_j according to

$$j = \arg\max_{n \in SV_{nb}} |O_i - O_n|. \tag{25}$$

(ii) Same as (i), but the scan is only performed over SV_{nb}.

In practice, one scan of type (i) is followed by multiple scans of type (ii), until there are no KKT violators in SV_{nb}, whereupon the optimization goes back to a single scan of type (i). If the type (i) scan finds no KKT violators, the optimization terminates.

In unusual circumstances, the choice heuristic (25) cannot make positive progress. Therefore, a hierarchy of other choice heuristics is applied to ensure positive progress.

[2]This scan can be accelerated by not checking patterns which are on the correct side of the hyperplane by a large margin, using the method of Joachims (1999).

These other heuristics are the same as in the case of pattern recognition, cf. (Platt, 1999), and have been found to work well in our experiments to be reported below.

In our experiments with SMO applied to distribution support estimation, we have always found it to converge. However, to ensure convergence even in rare pathological conditions, the algorithm can be modified slightly, cf. (Keerthi et al., 1999).

We end this section by stating a trick which is of importance in practical implementations. In practice, one has to use a nonzero accuracy tolerance when checking whether two quantities are equal. In particular, comparisons of this type are used in determining whether a point lies on the margin. Since we want the final decision function to evaluate to 1 for points which lie *on* the margin, we need to subtract this constant from the offset ρ at the end.

In the next section, it will be argued that subtracting something from ρ is actually advisable also from a statistical point of view.

5 Theory

We now analyse the algorithm theoretically, starting with the uniqueness of the hyperplane (Proposition 2). We then describe the connection to pattern recognition (Proposition 3), and show that the parameter ν characterizes the fractions of SVs and outliers (Proposition 4). Following that, we give a robustness result for the soft margin (Proposition 5) and finally we present a theoretical result on the generalization error (Theorem 7). The proofs are given in the Appendix.

In this section, we will use italic letters to denote the feature space images of the corresponding patterns in input space, i.e.

$$x_i := \Phi(\mathbf{x}_i). \tag{26}$$

Definition 1 *A data set*

$$x_1, \ldots, x_\ell \tag{27}$$

is called separable *if there exists some $w \in F$ such that $(w \cdot x_i) > 0$ for $i \in [\ell]$.*

If we use a Gaussian kernel (3), then any data set $\mathbf{x}_1, \ldots, \mathbf{x}_\ell$ is separable after it is mapped into feature space: to see this, note that $k(\mathbf{x}_i, \mathbf{x}_j) > 0$ for all i, j, thus all inner products between mapped patterns are positive, implying that all patterns lie inside the same orthant. Moreover, since $k(\mathbf{x}_i, \mathbf{x}_i) = 1$ for all i, they all have unit length. Hence they are separable from the origin.

Proposition 2 (Supporting Hyperplane) *If the data set (27) is separable, then there exists a unique* supporting hyperplane *with the properties that (1) it separates all data from the origin, and (2) its distance to the origin is maximal among all such hyperplanes. For any $\rho > 0$, it is given by*

$$\min_{w \in F} \frac{1}{2}\|w\|^2 \text{ subject to } (w \cdot x_i) \geq \rho, \quad i \in [\ell]. \tag{28}$$

The following result elucidates the relationship between single-class classification and binary classification.

Proposition 3 (Connection to Pattern Recognition) *(i) Suppose (w, ρ) parametrizes the supporting hyperplane for the data (27). Then $(w, 0)$ parametrizes the optimal separating hyperplane for the labelled data set*

$$\{(x_1, 1), \ldots, (x_\ell, 1), (-x_1, -1), \ldots, (-x_\ell, -1)\}. \tag{29}$$

(ii) Suppose $(w, 0)$ parametrizes the optimal separating hyperplane passing through the origin for a labelled data set

$$\{(x_1, y_1), \ldots, (x_\ell, y_\ell)\}, \quad (y_i \in \{\pm 1\} \text{ for } i \in [\ell]), \tag{30}$$

aligned such that $(w \cdot x_i)$ is positive for $y_i = 1$. Suppose, moreover, that $\rho/\|w\|$ is the margin of the optimal hyperplane. Then (w, ρ) parametrizes the supporting hyperplane for the unlabelled data set

$$\{y_1 x_1, \ldots, y_\ell x_\ell\}. \tag{31}$$

Note that the relationship is similar for nonseparable problems. In that case, *margin errors* in binary classification (i.e. points which are either on the wrong side of the separating hyperplane or which fall inside the margin) translate into *outliers* in single-class classification, i.e. into points which fall on the wrong side of the hyperplane. Proposition 3 then holds, cum grano salis, for the training sets with margin errors and outliers, respectively, removed.

The utility of Proposition 3 lies in the fact that it allows us to recycle certain results proven for binary classification (Schölkopf et al., 2000) for use in the single-class scenario. The following, explaining the significance of the parameter ν, is such a case.

Proposition 4 (ν-Property) *Assume the solution of (4),(5) satisfies $\rho \neq 0$. The following statements hold:*
(i) ν is an upper bound on the fraction of outliers.
(ii) ν is a lower bound on the fraction of SVs.
(iii) Suppose the data (27) were generated independently from a distribution $P(\mathbf{x})$ which does not contain discrete components. Suppose, moreover, that the kernel is analytic and non-constant. With probability 1, asymptotically, ν equals both the fraction of SVs and the fraction of outliers.

Note that this result also applies to the soft margin ball algorithm of Tax and Duin (1999), provided that it is stated in the ν-parameterization given in Sec. 3.

Proposition 5 (Resistance) *Local movements of outliers parallel to w do not change the hyperplane.*

Note that although the hyperplane does not change, its parametrization in w and ρ does.

We now move on to the subject of generalization. Our goal is to bound the probability that a novel point drawn from the same underlying distribution lies outside of the estimated region. We present a "marginalised" analysis which in fact provides a bound on the probability that a novel point lies outside the region slightly larger than the estimated one.

Definition 6 *Let f be a real-valued function on a space $\mathcal{X}$. Fix $\theta \in \mathbb{R}$. For $\mathbf{x} \in \mathcal{X}$ let $d(\mathbf{x}, f, \theta) = \max\{0, \theta - f(\mathbf{x})\}$. Similarly for a training sequence $\mathbf{X} := (\mathbf{x}_1, \ldots, \mathbf{x}_\ell)$, we define*

$$\mathcal{D}(\mathbf{X}, f, \theta) = \sum_{x \in \mathbf{X}} d(x, f, \theta).$$

In the following, $\log$ denotes logarithms to base 2 and $\ln$ denotes natural logarithms.

Theorem 7 (Generalization Error Bound) *Suppose we are given a set of ℓ examples $\mathbf{X} \in \mathcal{X}^\ell$ and we solve the optimisation problem (4),(5) (or equivalently (11)) and obtain a solution f_w given explicitly by (10). Let $R_{w,\rho} := \{\mathbf{x}: f_w(\mathbf{x}) \geq \rho\}$ denote the induced decision region. With probability $1 - \delta$ over the draw of the random sample $\mathbf{X} \in \mathcal{X}^\ell$, for any $\gamma > 0$,*

$$P\left\{\mathbf{x}': \mathbf{x}' \notin R_{w,\rho-\gamma}\right\} \leq \frac{2}{\ell}\left(k + \log\frac{\ell^2}{2\delta}\right), \tag{32}$$

where

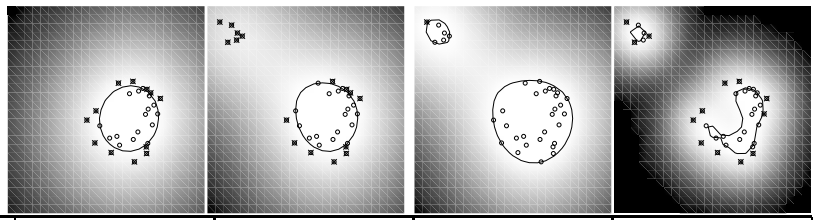
$$k = \frac{c_1 \log(c_2 \hat{\gamma}^2 \ell)}{\hat{\gamma}^2} + \frac{2\mathcal{D}}{\hat{\gamma}} \log\left(e\left(\frac{(2\ell-1)\hat{\gamma}}{2\mathcal{D}} + 1\right)\right) + 2, \tag{33}$$

$c_1 = 16c^2$, $c_2 = \ln(2)/(4c^2)$, $c = 103$, $\hat{\gamma} = \gamma/\|w\|$, $\mathcal{D} = \mathcal{D}(\mathbf{X}, f_{w,0}, \rho) = \mathcal{D}(\mathbf{X}, f_{w,\rho}, 0)$, and ρ is given by (12).

The training sample $\mathbf{X}$ defines (via the algorithm) the decision region $R_{w,\rho}$. We expect that new points generated according to P will lie in $R_{w,\rho}$. The theorem gives a probabilistic guarantee that new points lie in the larger region $R_{w,\rho-\gamma}$.

The parameter ν can be adjusted when running the algorithm to trade off incorporating outliers versus minimizing the "size" of $R_{w,\rho}$. Adjusting ν will change the value of $\mathcal{D}$. Note that since $\mathcal{D}$ is measured with respect to ρ while the bound applies to $\rho - \gamma$, any point which is outside of the region that the bound applies to will make a contribution to $\mathcal{D}$ which is bounded away from 0. Therefore, (32) does *not* imply that asymptotically, we will always estimate the complete support.

The parameter γ allows one to trade off the confidence with which one wishes the assertion of the theorem to hold against the size of the predictive region $R_{w,\rho-\gamma}$: one can see from (33) that k and hence the RHS of (32) scales inversely with γ. In fact, it scales inversely with $\hat{\gamma}$, i.e. it increases with w. This justifies measuring the complexity of the estimated region by the size of w, and minimizing $\|w\|^2$ in order to find a region that will generalize well. In addition, the theorem suggests not to use the threshold ρ returned by the algorithm, which would correspond to $\gamma = 0$, but a smaller value $\rho - \gamma$ (with $\gamma > 0$).



ν, width c	0.5, 0.5	0.5, 0.5	0.1, 0.5	0.5, 0.1
frac. SVs/OLs	0.54, 0.43	0.59, 0.47	0.24, 0.03	0.65, 0.38
margin $\rho/\|w\|$	0.84	0.70	0.62	0.48

Figure 1: *First two pictures:* A single-class SVM applied to two toy problems; $\nu = c = 0.5$, domain: $[-1, 1]^2$. Note how in both cases, at least a fraction of ν of all examples is in the estimated region (cf. table). The large value of ν causes the additional data points in the upper left corner to have almost no influence on the decision function. For smaller values of ν, such as 0.1 *(third picture)*, the points cannot be ignored anymore. Alternatively, one can force the algorithm to take these 'outliers' (OLs) into account by changing the kernel width (3): in the *fourth picture*, using $c = 0.1, \nu = 0.5$, the data is effectively analyzed on a different length scale which leads the algorithm to consider the outliers as meaningful points.

We do not claim that using Theorem 7 directly is a practical means to determine the parameters ν and γ explicitly. It is loose in several ways. We suspect c is too large by a factor of more than 50. Furthermore, no account is taken of the smoothness of the kernel used. If that were done (by using refined bounds on the covering numbers of the induced class of functions as in Williamson et al. (1998)), then the first term in (33) would increase much more slowly when decreasing γ. The fact that the second term would not change indicates a different tradeoff point. Nevertheless, the theorem gives one some confidence that ν and γ are suitable parameters to adjust.

6 Experiments

We apply the method to artificial and real-world data. Figure 1 displays 2-D toy examples, and shows how the parameter settings influence the solution. Figure 2 shows a comparison to a Parzen windows estimator on a 2-D problem, along with a family of estimators which lie "in between" the present one and the Parzen one.

Figure 3 shows a plot of the outputs $(w \cdot \Phi(\mathbf{x}))$ on training and test sets of the US postal service database of handwritten digits. The database contains 9298 digit images of size $16 \times 16 = 256$; the last 2007 constitute the test set. We used a Gaussian kernel (3), which has the advantage that the data are always separable from the origin in feature space (cf. the comment following Definition 1). For the kernel parameter c, we used $0.5 \cdot 256$. This value was chosen a priori, it is a common value for SVM classifiers on that data set, cf. Schölkopf et al. (1995)).[3] We fed our algorithm with the training

[3] In (Hayton et al., 2001), the following procedure is used to determine a value of c. For small c, all training points will become SVs — the algorithm just memorizes the data, and will not generalize well. As c increases, the number of SVs drops. As a simple heuristic, one can thus start with a small value of c and increase it until the number of SVs does not decrease any further.

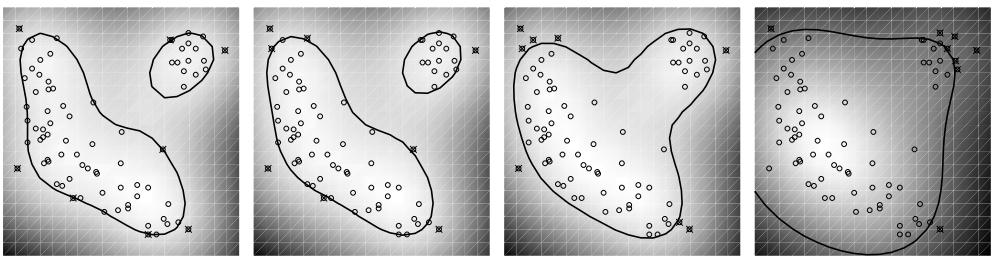


Figure 2: A single-class SVM applied to a toy problem; $c = 0.5$, domain: $[-1, 1]^2$, for various settings of the offset ρ. As discussed in Sec. 3, $\nu = 1$ yields a Parzen windows expansion. However, to get a Parzen windows estimator of the distribution's support, we must in that case not use the threshold returned by the algorithm (which would allow *all* points to lie outside of the estimated region). Therefore, in this experiment, we adjusted the threshold such that a fraction $\nu' = 0.1$ of patterns would lie outside. From left to right, we show the results for $\nu \in \{0.1, 0.2, 0.4, 1\}$. The rightmost picture corresponds to the Parzen estimator which utilizes all kernels; the other estimators use roughly a fraction of ν kernels. Note that as a result of the averaging over all kernels, the Parzen windows estimate does not model the shape of the distribution very well for the chosen parameters.

instances of digit 0 only. Testing was done on both digit 0 and on all other digits. We present results for two values of ν, one large, one small; for values in between, the results are qualitatively similar. In the first experiment, we used $\nu = 50\%$, thus aiming for a description of "0-ness" which only captures half of all zeros in the training set. As shown in figure 3, this leads to *zero* false positives (i.e. even though the learning machine has not seen any non-0-s during training, it correctly identifies all non-0-s as such), while still recognizing 44% of the digits 0 in the *test* set. Higher recognition rates can be achieved using smaller values of ν: for $\nu = 5\%$, we get 91% correct recognition of digits 0 in the test set, with a fairly moderate false positive rate of 7%.

Whilst leading to encouraging results, this experiment did not really address the actual task the algorithm was designed for. Therefore, we next focussed on a problem of novelty detection. Again, we utilized the USPS set; however, this time we trained the algorithm on the test set and used it to identify outliers — it is folklore in the community that the USPS test set (Fig. 4) contains a number of patterns which are hard or impossible to classify, due to segmentation errors or mislabelling (e.g. Vapnik, 1995). In this experiment, we augmented the input patterns by ten extra dimensions corresponding to the class labels of the digits. The rationale for this is that if we disregarded the labels, there would be no hope to identify mislabelled patterns as outliers. With the labels, the algorithm has the chance to identify both unusual patterns and usual patterns with unusual labels. Fig. 5 shows the 20 worst outliers for the USPS test set, respectively. Note that the algorithm indeed extracts patterns which are very hard to assign to their respective classes. In the experiment, we used the same kernel width as above, and a ν value of 5%. The latter was chosen roughly to reflect our expectations as to how many "bad" patterns there are in the test set: most good learning algorithms achieve error rates of 3 - 5% on the USPS benchmark (for a list of results, cf. Vapnik (1995)).

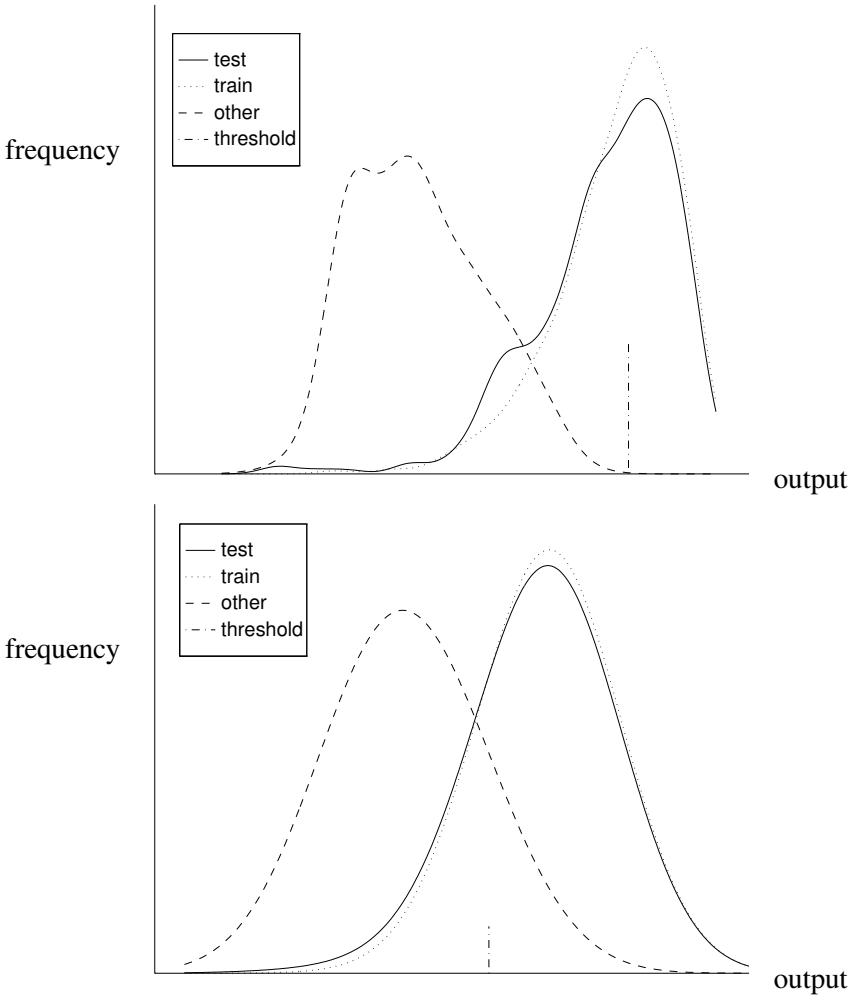


Figure 3: Experiments on the US postal service OCR dataset. Recognizer for digit 0; output histogram for the exemplars of 0 in the training/test set, and on test exemplars of other digits. The x-axis gives the output values, i.e. the argument of the sgn function in (10). For $\nu = 50\%$ *(top)*, we get 50% SVs and 49% outliers (consistent with Proposition 4), 44% true positive test examples, and zero false positives from the "other" class. For $\nu = 5\%$ *(bottom)*, we get 6% and 4% for SVs and outliers, respectively. In that case, the true positive rate is improved to 91%, while the false positive rate increases to 7%. The threshold ρ is marked in the graphs.
Note, finally, that the plots show a Parzen windows density estimate of the output histograms. In reality, many examples sit exactly at the threshold value (the non-bound SVs). Since this peak is smoothed out by the estimator, the fractions of outliers in the training set appear slightly larger than it should be.

ν	fraction of OLs	fraction of SVs	training time
1%	0.0%	10.0%	36
2%	0.0%	10.0%	39
3%	0.1%	10.0%	31
4%	0.6%	10.1%	40
5%	**1.4%**	**10.6%**	**36**
6%	1.8%	11.2%	33
7%	2.6%	11.5%	42
8%	4.1%	12.0%	53
9%	5.4%	12.9%	76
10%	6.2%	13.7%	65
20%	16.9%	22.6%	193
30%	27.5%	31.8%	269
40%	37.1%	41.7%	685
50%	47.4%	51.2%	1284
60%	58.2%	61.0%	1150
70%	68.3%	70.7%	1512
80%	78.5%	80.5%	2206
90%	89.4%	90.1%	2349

Table 1: Experimental results for various values of the outlier control constant ν, USPS test set, size $\ell = 2007$. Note that ν bounds the fractions of outliers and support vectors from above and below, respectively (cf. Proposition 4). As we are not in the asymptotic regime, there is some slack in the bounds; nevertheless, ν can be used to control the above fractions. Note, moreover, that training times (CPU time in seconds on a Pentium II running at 450 MHz) increase as ν approaches 1. This is related to the fact that almost all Lagrange multipliers will be at the upper bound in that case (cf. Sec. 4). The system used in the outlier detection experiments is shown in bold face.

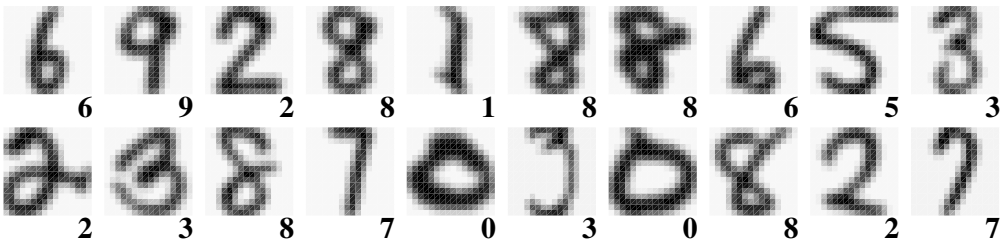


Figure 4: A subset of 20 examples randomly drawn from the USPS test set, with class labels.

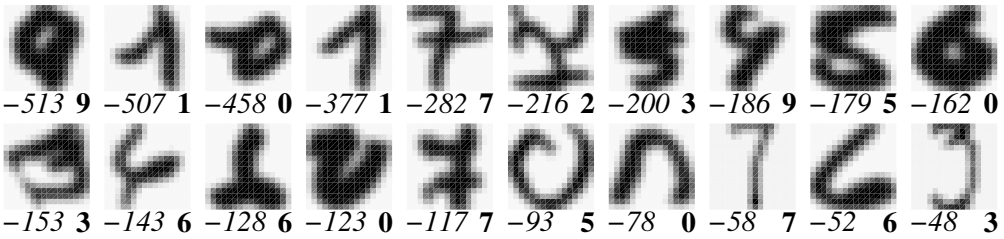


Figure 5: Outliers identified by the proposed algorithm, ranked by the negative output of the SVM (the argument of (10)). The outputs (for convenience in units of 10^{-5}) are written underneath each image in italics, the (alleged) class labels are given in bold face. Note that most of the examples are "difficult" in that they are either atypical or even mislabelled.

In the last experiment, we tested the runtime scaling behaviour of the proposed SMO solver which is used for training the learning machine (Fig. 6). It was found to depend on the value of ν utilized. For the small values of ν which are typically used in outlier detection tasks, the algorithm scales very well to larger data sets, with a dependency of training times on the sample size which is at most quadratic.

In addition to the experiments reported above, the present algorithm has since been applied in several other domains, such as the modelling of parameter regimes for the control of walking robots (Still and Schölkopf, 2001), and condition monitoring of jet engines (Hayton et al., 2001).

7 Discussion

One could view the present work as an attempt to provide a new algorithm which is in line with Vapnik's principle never to solve a problem which is more general than the one that one is actually interested in. E.g., in situations where one is only interested in detecting *novelty*, it is not always necessary to estimate a full density model of the data. Indeed, density estimation is more difficult than what we are doing, in several respects.

Mathematically speaking, a density will only exist if the underlying probability measure possesses an absolutely continuous distribution function. However, the general problem of estimating the measure for a large class of sets, say the sets measureable in Borel's sense, is not solvable (for a discussion, see e.g. Vapnik, 1998). Therefore we need to restrict ourselves to making a statement about the measure of *some* sets. Given a small class of sets, the simplest estimator which accomplishes this task is the empirical measure, which simply looks at how many training points fall into the region of interest. Our algorithm does the opposite. It starts with the number of training points that are supposed to fall into the region, and then estimates a region with the desired property. Often, there will be many such regions — the solution becomes unique only by applying a regularizer, which in our case enforces that the region be small in a feature space associated to the kernel.

Therefore, we must keep in mind that the measure of smallness in this sense de-

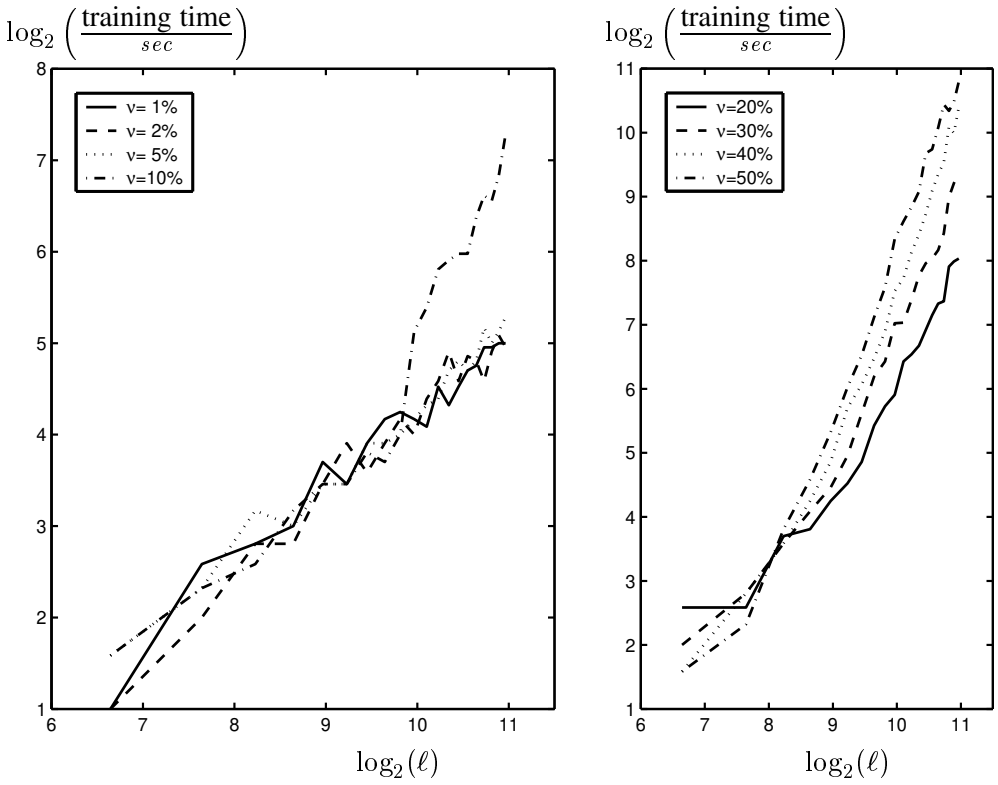


Figure 6: Training times vs. data set sizes ℓ (both axes depict logs at base 2; CPU time in seconds on a Pentium II running at 450 MHz, training on subsets of the USPS test set); $c = 0.5 \cdot 256$. As in Table 1, it can be seen that larger values of ν generally lead to longer training times (note that the plots use different y-axis ranges). However, they also differ in their scaling with the sample size. The exponents can be directly read off from the slope of the graphs, as they are plotted in log scale with equal axis spacing: for small values of ν ($\leq 5\%$), the training times were approximately linear in the training set size. The scaling gets worse as ν increases. For large values of ν, training times are roughly proportional to the sample size raised to the power of 2.5 (right plot). The results should be taken only as an indication of what is going on: they were obtained using fairly small training sets, the largest being 2007, the size of the USPS test set. As a consequence, they are fairly noisy, and they strictly speaking only refer to the examined regime. Encouragingly, the scaling is better than the cubic one that one would expect when solving the optimization problem using all patterns at once, cf. Sec. 4. Moreover, for *small* values of ν, i.e. those typically used in outlier detection (in Fig. 5, we used $\nu = 5\%$), our algorithm was particularly efficient.

pends on the kernel used, in a way that is no different to any other method that regularizes in a feature space. A similar problem, however, appears in density estimation already when done in input space. Let p denote a density on $\mathcal{X}$. If we perform a (nonlinear) coordinate transformation in the input domain $\mathcal{X}$, then the density values will *change*; loosely speaking, what remains constant is $p(\mathbf{x}) \cdot d\mathbf{x}$, while $d\mathbf{x}$ is transformed, too. When directly estimating the probability *measure* of regions, we are not faced with this problem, as the regions automatically change accordingly.

An attractive property of the measure of smallness that we chose to use is that it can also be placed in the context of regularization theory, leading to an interpretation of the solution as maximally smooth in a sense which depends on the specific kernel used. More specifically, let us assume that k is Green's function of P^*P for an operator P mapping into some inner product space (Smola et al., 1998; Girosi, 1998), and take a look at the dual objective function that we minimize,

$$\begin{aligned}
\sum_{i,j} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) &= \sum_{i,j} \alpha_i \alpha_j (k(\mathbf{x}_i, .) \cdot \delta_{\mathbf{x}_j}(.)) \\
&= \sum_{i,j} \alpha_i \alpha_j (k(\mathbf{x}_i, .) \cdot (P^*Pk)(\mathbf{x}_j, .)) \\
&= \sum_{i,j} \alpha_i \alpha_j ((Pk)(\mathbf{x}_i, .) \cdot (Pk)(\mathbf{x}_j, .)) \\
&= ((P \sum_i \alpha_i k)(\mathbf{x}_i, .) \cdot (P \sum_j \alpha_j k)(\mathbf{x}_j, .)) \\
&= \|Pf\|^2,
\end{aligned}$$

using $f(\mathbf{x}) = \sum_i \alpha_i k(\mathbf{x}_i, \mathbf{x})$. The regularization operators of common kernels can be shown to correspond to derivative operators (Poggio and Girosi, 1990) — therefore, minimizing the dual objective function corresponds to maximizing the smoothness of the function f (which is, up to a thresholding operation, the function we estimate). This, in turn, is related to a prior $p(f) \sim e^{-\|Pf\|^2}$ on the function space.

Interestingly, as the minimization of the dual objective function also corresponds to a maximization of the margin in feature space, an equivalent interpretation is in terms of a prior on the distribution of the unknown other class (the "novel" class in a novelty detection problem) — trying to separate the data from the origin amounts to assuming that the novel examples lie around the origin.

The main inspiration for our approach stems from the earliest work of Vapnik and collaborators. In 1962, they proposed an algorithm for characterizing a set of unlabelled data points by separating it from the origin using a hyperplane (Vapnik and Lerner, 1963; Vapnik and Chervonenkis, 1974). However, they quickly moved on to two-class classification problems, both in terms of algorithms and in terms of the theoretical development of statistical learning theory which originated in those days.

From an algorithmic point of view, we can identify two shortcomings of the original approach which may have caused research in this direction to stop for more than three decades. Firstly, the original algorithm in (Vapnik and Chervonenkis, 1974) was limited to linear decision rules in input space; secondly, there was no way of dealing with outliers. In conjunction, these restrictions are indeed severe — a generic dataset need not be separable from the origin by a hyperplane in input space.

The two modifications that we have incorporated dispose of these shortcomings. First, the kernel trick allows for a much larger class of functions by nonlinearly mapping into a high-dimensional feature space, and thereby increases the chances of a separation from the origin being possible. In particular, using a Gaussian kernel (3), such a separation is always possible, as shown in Sec. 5. The second modification directly allows for the possibility of outliers. We have incorporated this 'softness' of the decision rule using the ν-trick (Schölkopf et al., 2000) and thus obtained a direct handle on the fraction of outliers.

We believe that our approach, proposing a concrete algorithm with well-behaved computational complexity (convex quadratic programming) for a problem that so far has mainly been studied from a theoretical point of view has abundant practical applications. To turn the algorithm into an easy-to-use black-box method for practicioners, questions like the selection of kernel parameters (such as the width of a Gaussian kernel) have to be tackled. It is our hope that the theoretical results which we have briefly outlined in this paper will provide a solid foundation for this formidable task. This, alongside with algorithmic extensions such as the possibility of taking into account information about the "abnormal" class Schölkopf et al. (2000), is subject of current research.

Acknowledgement. This work was supported by the ARC, the DFG (# Ja 379/9-1 and Sm 62/1-1), and by the European Commission under the Working Group Nr. 27150 (NeuroCOLT2). Parts of it were done while BS and AS were with GMD FIRST, Berlin. Thanks to S. Ben-David, C. Bishop, P. Hayton, N. Oliver, C. Schnörr, J. Spanjaard, L. Tarassenko, and M. Tipping for helpful discussions.

References

S. Ben-David and M. Lindenbaum. Learning distributions by their density levels: A paradigm for learning without a teacher. *Journal of Computer and System Sciences*, 55:171–182, 1997.

D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 1995.

B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152, Pittsburgh, PA, 1992. ACM Press.

J. Chevalier. Estimation du support et du contour du support d'une loi de probabilité. *Annales de l'Institut Henri Poincaré. Section B. Calcul des Probabilités et Statistique. Nouvelle Série*, 12(4):339–364, 1976.

T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley, New York, 1991.

A. Cuevas. On pattern analysis in the non-convex case. *Kybernetes*, 19(6):26–33, 1990.

A. Cuevas and R. Fraiman. A plug-in approach to support estimation. *The Annals of Statistics*, 25(6):2300–2312, 1997.

L. Devroye and G.L. Wise. Detection of abnormal behaviour via nonparametric estimation of the support. *SIAM Journal on Applied Mathematics*, 38(3):480–488, 1980.

J.H.J. Einmal and D.M. Mason. Generalized quantile processes. *The Annals of Statistics*, 20(2):1062–1078, 1992.

G. Gayraud. Estimation of functional of density support. *Mathematical Methods of Statistics*, 6(1):26–46, 1997.

J. Geffroy. Sur un problème d'estimation géométrique. *Publications de l'Institut de Statistique de l'Université de Paris*, 13:191–210, 1964.

F. Girosi. An equivalence between sparse approximation and support vector machines. *Neural Computation*, 10(6):1455–1480, 1998.

J.A. Hartigan. Estimation of a convex density contour in two dimensions. *Journal of the American Statistical Association*, 82:267–270, 1987.

P. Hayton, B. Schölkopf, L. Tarassenko, and P. Anuzis. Support vector novelty detection applied to jet engine vibration spectra. In *Advances in Neural Information Processing Systems 13*, 2001. Forthcoming.

T. Joachims. Making large–scale SVM learning practical. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods — Support Vector Learning*, pages 169–184, Cambridge, MA, 1999. MIT Press.

S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, and K.R.K. Murthy. Improvements to Platt's SMO algorithm for SVM classifier design. Technical Report CD-99-14, Dept. of Mechanical and Production Engineering, Natl. Univ. Singapore, Singapore, 1999.

A.P. Korostelev and A.B. Tsybakov. *Minimax Theory of Image Reconstruction*. Springer, New York, 1993.

D.W. Müller. The excess mass approach in statistics. Beiträge zur Statistik, Universität Heidelberg, 1992.

D. Nolan. The excess mass ellipsoid. *Journal of Multivariate Analysis*, 39:348–371, 1991.

J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods — Support Vector Learning*, pages 185–208, Cambridge, MA, 1999. MIT Press.

T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9), 1990.

W. Polonik. Density estimation under qualitative assumptions in higher dimensions. *Journal of Multivariate Analysis*, 55(1):61–81, 1995a.

W. Polonik. Measuring mass concentrations and estimating density contour clusters — an excess mass approach. *The Annals of Statistics*, 23(3):855–881, 1995b.

W. Polonik. Minimum volume sets and generalized quantile processes. *Stochastic Processes and their Applications*, 69:1–24, 1997.

T. W. Sager. An iterative method for estimating a multivariate mode and isopleth. *Journal of the American Statistical Association*, 74(366):329–339, 1979.

B. Schölkopf, C. Burges, and V. Vapnik. Extracting support data for a given task. In U. M. Fayyad and R. Uthurusamy, editors, *Proceedings, First International Conference on Knowledge Discovery & Data Mining*, Menlo Park, 1995. AAAI Press.

B. Schölkopf, C. J. C. Burges, and A. J. Smola. *Advances in Kernel Methods — Support Vector Learning*. MIT Press, Cambridge, MA, 1999a.

B. Schölkopf, J. Platt, and A.J. Smola. Kernel method for percentile feature extraction. TR MSR 2000 - 22, Microsoft Research, Redmond, WA, 2000.

B. Schölkopf, A. Smola, and K.-R. Müller. Kernel principal component analysis. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 327–352. MIT Press, Cambridge, MA, 1999b.

B. Schölkopf, A. Smola, R. C. Williamson, and P. L. Bartlett. New support vector algorithms. *Neural Computation*, 12:1207 – 1245, 2000.

B. Schölkopf, R. Williamson, A. Smola, and J. Shawe-Taylor. Single-class support vector machines. In J. Buhmann, W. Maass, H. Ritter, and N. Tishby, editors, *Unsupervised Learning*, Dagstuhl-Seminar-Report 235, pages 19 – 20, 1999.

J. Shawe-Taylor and N. Cristianini. Margin distribution bounds on generalization. In *Computational Learning Theory: 4th European Conference*, volume 1572 of *Lecture Notes in Artificial Intelligence*, pages 263 – 273. Springer, 1999.

J. Shawe-Taylor and N. Cristianini. On the generalisation of soft margin algorithms. *IEEE Transactions on Information Theory*, 2000. Submitted.

J. Shawe-Taylor and R. C. Williamson. Generalization performance of classifiers in terms of observed covering numbers. In *Proc. EUROCOLT'99*, 1999.

A. Smola, S. Mika, B. Schölkopf, and R. C. Williamson. Regularized principal manifolds. *Machine Learning*, 2000. Forthcoming.

A. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 2000. Forthcoming.

A. Smola, B. Schölkopf, and K.-R. Müller. The connection between regularization operators and support vector kernels. *Neural Networks*, 11:637–649, 1998.

S. Still and B. Schölkopf. Four-legged walking gait control using a neuromorphic chip interfaced to a support vector learning algorithm. In *Advances in Neural Information Processing Systems 13*, 2001. Forthcoming.

D. Stoneking. Improving the manufacturability of electronic designs. *IEEE Spectrum*, 36(6):70–76, 1999.

L. Tarassenko, P. Hayton, N. Cerneaz, and M. Brady. Novelty detection for the identification of masses in mammograms. In *Proceedings Fourth IEE International Conference on Artificial Neural Networks*, pages 442 – 447, Cambridge, 1995.

D.M.J. Tax and R.P.W. Duin. Data domain description by support vectors. In M. Verleysen, editor, *Proceedings ESANN*, pages 251 – 256, Brussels, 1999. D Facto.

A.B. Tsybakov. On nonparametric estimation of density level sets. *The Annals of Statistics*, 25(3):948–969, 1997.

V. Vapnik. *Estimation of Dependences Based on Empirical Data [in Russian]*. Nauka, Moscow, 1979. (English translation: Springer Verlag, New York, 1982).

V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, N.Y., 1995.

V. Vapnik. *Statistical Learning Theory*. Wiley, N.Y., 1998.

V. Vapnik and A. Chervonenkis. *Theory of Pattern Recognition [in Russian]*. Nauka, Moscow, 1974. (German Translation: W. Wapnik & A. Tscherwonenkis, *Theorie der Zeichenerkennung*, Akademie–Verlag, Berlin, 1979).

V. Vapnik and A. Lerner. Pattern recognition using generalized portrait method. *Automation and Remote Control*, 24, 1963.

R. C. Williamson, A. J. Smola, and B. Schölkopf. Generalization performance of regularization networks and support vector machines via entropy numbers of compact operators. Technical Report 19, NeuroCOLT, http://www.neurocolt.com, 1998. Accepted for publication in IEEE Transactions on Information Theory.

R. C. Williamson, A. J. Smola, and B. Schölkopf. Entropy numbers of linear function classes. In N. Cesa-Bianchi and S. Goldman, editors, *Proceedings of the 13th Annual Conference on Computational Learning Theory*, pages 309 – 319, San Francisco, 2000. Morgan Kaufman.

A Supplementary Material for Sec. 2

Estimating the support of a density. The problem of estimating $C(1)$ appears to have first been studied by Geffroy (1964) who considered $\mathcal{X} = \mathbb{R}^2$ with piecewise constant estimators. There have been a number of works studying a natural nonparametric estimator of $C(1)$ (e.g. Chevalier (1976); Devroye and Wise (1980); see (Gayraud, 1997) for further references). The nonparametric estimator is simply

$$\hat{C}_\ell = \bigcup_{i=1}^{\ell} B(\mathbf{x}_i, \epsilon_\ell) \tag{34}$$

where $B(\mathbf{x}, \epsilon)$ is the $l_2(\mathcal{X})$ ball of radius ϵ centered at $\mathbf{x}$ and $(\epsilon_\ell)_\ell$ is an appropriately chosen decreasing sequence. Devroye and Wise (1980) showed the asymptotic consistency of (34) with respect to the symmetric difference between $C(1)$ and $\hat{C}_\ell$. Cuevas (1990) did the same, but for Hausdorff distance. Cuevas and Fraiman (1997) studied the asymptotic consistency of a *plug-in* estimator of $C(1)$: $\hat{C}^{\mathrm{plug-in}} = \{\mathbf{x}\colon \hat{p}_\ell(\mathbf{x}) > 0\}$ where $\hat{p}_\ell$ is a kernel density estimator. In order to avoid problems with $\hat{C}^{\mathrm{plug-in}}$ they actually analyzed $\hat{C}_\beta^{\mathrm{plug-in}} := \{\mathbf{x}\colon \hat{p}_\ell(\mathbf{x}) > \beta_\ell\}$ where $(\beta_\ell)_\ell$ is an appropriately chosen sequence. Clearly for a given distribution, α is related to β, but this connection can not be readily exploited by this type of estimator.

The most recent work relating to the estimation of $C(1)$ is by Gayraud (1997) who has made an asymptotic minimax study of estimators of *functionals* of $C(1)$. Two examples are $\mathrm{vol}\, C(1)$ or the center of $C(1)$. (See also (Korostelev and Tsybakov, 1993, Chapter 8).)

Estimating high probability regions ($\alpha \neq 1$). Polonik (1995b) has studied the use of the "excess mass approach" (Müller, 1992) to construct an estimator of "generalized α-clusters" which are related to $C(\alpha)$.

Define the *excess mass over $\mathcal{C}$ at level α* as

$$E_{\mathcal{C}}(\alpha) = \sup\{H_\alpha(C)\colon C \in \mathcal{C}\}$$

where $H_\alpha(\cdot) = (P - \alpha\lambda)(\cdot)$ and again λ denotes Lebesgue measure. Any set $\Gamma_{\mathcal{C}}(\alpha) \in \mathcal{C}$ such that

$$E_{\mathcal{C}}(\alpha) = H_\alpha(\Gamma_{\mathcal{C}}(\alpha))$$

is called a *generalized α-cluster in $\mathcal{C}$*. Replace P by P_ℓ in these definitions to obtain their empirical counterparts $E_{\ell,\mathcal{C}}(\alpha)$ and $\Gamma_{\ell,\mathcal{C}}(\alpha)$. In other words, his estimator is

$$\Gamma_{\ell,\mathcal{C}}(\alpha) = \arg\max\{(P_\ell - \alpha\lambda)(C) : C \in \mathcal{C}\}$$

where the max is not necessarily unique. Now whilst $\Gamma_{\ell,\mathcal{C}}(\alpha)$ is clearly different from $C_\ell(\alpha)$, it is related to it in that it attempts to find small regions with as much excess mass (which is similar to finding small regions with a given amount of probability mass). Actually $\Gamma_{\ell,\mathcal{C}}(\alpha)$ is more closely related to the determination of *density contour clusters* at level α:

$$c_p(\alpha) := \{\mathbf{x}\colon p(\mathbf{x}) \geq \alpha\}.$$

Simultaneously, and independently, Ben-David and Lindenbaum (1997) studied the problem of estimating $c_p(\alpha)$. They too made use of VC classes but stated their results in a stronger form which is meaningful for finite sample sizes.

Finally we point out a curious connection between minimum volume sets of a distribution and its differential entropy in the case that $\mathcal{X}$ is one dimensional. Suppose X is a one dimensional random variable with density p. Let $S = C(1)$ be the support of p and define the *differential entropy* of X by $h(X) = -\int_S p(x)\log p(x)dx$. For $\epsilon > 0$ and $\ell \in \mathbb{N}$, define the *typical set* $A_\epsilon^{(\ell)}$ with respect to p by

$$A_\epsilon^{(\ell)} = \{(x_1, \ldots, x_\ell) \in S^\ell\colon |-\tfrac{1}{\ell}\log p(x_1, \ldots, x_\ell) - h(X)| \leq \epsilon\},$$

where $p(x_1, \ldots, x_\ell) = \prod_{i=1}^{\ell} p(x_i)$. If $(a_\ell)_\ell$ and $(b_\ell)_\ell$ are sequences, the notation $a_\ell \doteq b_\ell$ means $\lim_{\ell\to\infty} \frac{1}{\ell} \log \frac{a_\ell}{b_\ell} = 0$. (Cover and Thomas, 1991, p.227) show that for all $\epsilon, \delta < \frac{1}{2}$, then

$$\mathrm{vol}\, A_\epsilon^{(\ell)} \doteq \mathrm{vol}\, C_\ell(1-\delta) \doteq 2^{\ell h}.$$

They point out that this result "indicates that the volume of the smallest set that contains most of the probability is approximately $2^{\ell h}$. This is a ℓ-dimensional volume, so the corresponding side length is $(2^{\ell h})^{1/\ell} = 2^h$. This provides an interpretation of differential entropy."

B Proofs of Sec. 5

Proof (Proposition 2) Due to the separability, the convex hull of the data does not contain the origin. The existence and uniqueness of the hyperplane then follows from the supporting hyperplane theorem (e.g. Bertsekas, 1995).

Moreover, separability implies that there actually exists some $\rho > 0$ and $w \in F$ such that $(w \cdot x_i) \geq \rho$ for $i \in [\ell]$ (by rescaling w, this can be seen to work for arbitrarily large ρ). The distance of the hyperplane $\{z \in F : (w \cdot z) = \rho\}$ to the origin is $\rho/\|w\|$. Therefore the optimal hyperplane is obtained by minimizing $\|w\|$ subject to these constraints, i.e. by the solution of (28). ∎

Proof (Proposition 3) *Ad (i).* By construction, the separation of (31) is a point-symmetric problem. Hence, the optimal separating hyperplane passes through the origin, for, if it did not, we could obtain another optimal separating hyperplane by reflecting the first one with respect to the origin — this would contradict the uniqueness of the optimal separating hyperplane Vapnik (1995).

Next, observe that $(-w, \rho)$ parametrizes the supporting hyperplane for the data set reflected through the origin, and that it is parallel to the one given by (w, ρ). This provides an optimal separation of the two sets, with distance 2ρ, and a separating hyperplane $(w, 0)$.

Ad (ii). By assumption, w is the shortest vector satisfying $y_i(w \cdot x_i) \geq \rho$ (note that the offset is 0). Hence, equivalently, it also the shortest vector satisfying $(w \cdot y_i x_i) \geq \rho$ for $i \in [\ell]$. ∎

Proof (Proposition 4) Parts (i) and (ii) follow directly from Proposition 3 and the fact that outliers are dealt with in exactly the same way as margin errors in the optimization problem for the binary classification case (Schölkopf et al., 2000). The basic idea is that (9) imposes constraints on the fraction of patterns that can have $\alpha_i = 1/(\nu\ell)$, upper bounding the fraction of outliers, and on the fraction of patterns that must have $\alpha_i > 0$, the SVs. Alternatively, the result can be proven directly based on the primal objective function (4), as sketched presently: to this end, note that when changing ρ, the term $\sum_i \xi_i$ will change proportionally to the *number* of points that have a nonzero ξ_i (the outliers), plus, when changing ρ in the positive direction, the number of points which are just about to get a nonzero ρ, i.e. which sit *on* the hyperplane (the SVs). At the optimum of (4), we therefore have (i) and (ii).

Part (iii) can be proven by a uniform convergence argument showing that since the covering numbers of kernel expansions regularized by a norm in some feature space are well-behaved, the fraction of points which lie exactly on the hyperplane is asymptotically negligible (cf. Schölkopf et al., 2000). ∎

Proof (Proposition 5) Suppose x_o is an outlier, i.e. $\xi_o > 0$, hence by the KKT conditions (e.g. Bertsekas, 1995) $\alpha_o = 1/(\nu\ell)$. Transforming it into $x'_o := x_o + \delta \cdot w$, where $|\delta| < \xi_o/\|w\|$, leads to a slack which is still nonzero, i.e. $\xi'_o > 0$, hence we still have $\alpha_o = 1/(\nu\ell)$. Therefore, $\boldsymbol{\alpha}' = \boldsymbol{\alpha}$ is still feasible, as is the primal solution $(w', \boldsymbol{\xi}', \rho')$. Here, we use $\xi'_i = (1 + \delta \cdot \alpha_o)\xi_i$ for $i \neq o$, $w' = (1 + \delta \cdot \alpha_o)w$, and ρ' as computed from (12). Finally, the KKT conditions are still satisfied, as still $\alpha'_o = 1/(\nu\ell)$. Thus (Bertsekas, 1995, e.g.), $\boldsymbol{\alpha}$ is still the optimal solution. ∎

We now move on to the Proof of Theorem 7. We start by introducing a common tool for measuring the capacity of a class $\mathcal{F}$ of functions that map $\mathcal{X}$ to $\mathbb{R}$.

Definition 8 *Let (X, d) be a pseudo-metric space, let A be a subset of X and $\epsilon > 0$. A set $U \subseteq X$ is an* ϵ-cover *for A if, for every $a \in A$, there exists $u \in U$ such that $d(a, u) \leq \epsilon$. The* ϵ-covering number *of A, $\mathcal{N}(\epsilon, A, d)$, is the minimal cardinality of an ϵ-cover for A (if there is no such finite cover then it is defined to be ∞).*

Note that we have used less than or equal to in the definition of a cover. This is somewhat unconventional, but will not change the bounds we use. It is, however, technically useful in the proofs.

The idea is that U should be finite but approximate all of A with respect to the pseudometric d. We will use the l_∞ norm over a finite sample $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_\ell)$ for the pseudo-norm on $\mathcal{F}$,

$$\|f\|_{l_\infty^{\mathbf{X}}} := \max_{\mathbf{x} \in \mathbf{X}} |f(\mathbf{x})|. \tag{35}$$

(The (pseudo)-norm induces a (pseudo)-metric in the usual way.) Suppose $\mathcal{X}$ is a compact subset of X. Let $\mathcal{N}(\epsilon, \mathcal{F}, \ell) = \max_{\mathbf{X} \in \mathcal{X}^\ell} \mathcal{N}(\epsilon, \mathcal{F}, l_\infty^{\mathbf{X}})$ (the maximum exists by definition of the covering number and compactness of $\mathcal{X}$).

We require a technical restriction on the function class which is satisfied by standard classes such as kernel based linear function classes and neural networks with bounded weights (see Shawe-Taylor and Williamson (1999) for further details.)

Definition 9 *Let*

$$S_{\mathbf{X}} : \mathcal{F} \to \mathbb{R}^\ell, \quad S_{\mathbf{X}} : f \mapsto (f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_\ell))$$

denote the multiple evaluation map induced by $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_\ell) \in \mathcal{X}^\ell$. We say that a class of functions $\mathcal{F}$ is sturdy *if for all $\ell \in \mathbb{N}$ and all $\mathbf{X} \in \mathcal{X}^\ell$ the image $S_{\mathbf{X}}(\mathcal{F})$ of $\mathcal{F}$ under $S_{\mathbf{X}}$ is a compact subset of $\mathbb{R}^\ell$.*

Lemma 10 (Shawe-Taylor and Williamson (1999)) *Let $\mathcal{F}$ be a sturdy class of functions. Then for each $N \in \mathbb{N}$, $N \geq 1$ and any fixed sequence $\mathbf{X} \in \mathcal{X}^\ell$, the infimum $\inf\{\gamma : \mathcal{N}(\gamma, \mathcal{F}, \mathbf{X}) \leq N\}$, is attained.*

We treat $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_\ell)$ both as a sequence and a set, for example writing $\mathbf{x} \in \mathbf{X}$ to mean $\mathbf{x} = \mathbf{x}_i$ for some $i \in [\ell]$. The concatenation of two sequences $\mathbf{X} = (\mathbf{x}_i)_{i\in[l]}$ and $\mathbf{X}' = (\mathbf{x}'_i)_{i\in[\ell]}$ is $\mathbf{X}\mathbf{X}' = (\mathbf{x}_1, \ldots, \mathbf{x}_\ell, \mathbf{x}'_1, \ldots, \mathbf{x}'_\ell)$.

We will make use of the following fundamental lemma.

Lemma 11 (Vapnik (1979), page 168) *Let $\mathcal{X}$ be a set and S a system of sets in $\mathcal{X}$, and P a probability measure on S. For $\mathbf{X} \in \mathcal{X}^\ell$ and $A \in S$, define $\nu_{\mathbf{X}}(A) := |\mathbf{X} \cap A|/\ell$. If $\ell > 2/\epsilon$, then*

$$P^\ell \left\{ \mathbf{X}: \sup_{A\in S} |\nu_{\mathbf{X}}(A) - P(A)| > \epsilon \right\} \leq 2P^{2\ell} \left\{ \mathbf{X}\mathbf{X}': \sup_{A\in S} |\nu_{\mathbf{X}}(A) - \nu_{\mathbf{X}'}(A)| > \epsilon/2 \right\}.$$

Below, the notations $\lceil t \rceil$ and $\lfloor t \rfloor$ denote the smallest integer $\geq t$ and the largest integer $\leq t$, respectively. Several of the theorems below claim some event occurs with probability $1 - \delta$. In all cases, $0 < \delta < 1$ is understood, and δ is presumed in fact to be small.

Theorem 12 *Consider any distribution P on $\mathcal{X}$ and a sturdy real-valued function class $\mathcal{F}$ on $\mathcal{X}$. Suppose $\mathbf{x}_1, \ldots, \mathbf{x}_\ell$ are generated i.i.d. from P. Then with probability $1 - \delta$ over such an ℓ-sample, for any $f \in \mathcal{F}$ and for any $\gamma > 0$,*

$$P\left\{ \mathbf{x} : f(\mathbf{x}) < \min_{i\in[\ell]} f(\mathbf{x}_i) - 2\gamma \right\} \leq \epsilon(\ell, k, \delta) := \tfrac{2}{\ell}(k + \log \tfrac{\ell}{\delta}),$$

where $k = \lceil \log \mathcal{N}(\gamma, \mathcal{F}, 2\ell) \rceil$.

Proof Fix $\gamma > 0$ and consider the event that the sample drawn fails to satisfy the criterion

$$J = \left\{ \mathbf{X} \in \mathcal{X}^\ell : \exists f \in \mathcal{F}, P\left\{ \mathbf{x} : f(\mathbf{x}) < \min_{i\in[\ell]} f(\mathbf{x}_i) - 2\gamma \right\} > \epsilon \right\}.$$

We must show that $P^\ell(J) \leq \delta$ for $\epsilon = \epsilon(\ell, k, \delta)$. We first consider a fixed k and apply Lemma 11 (note that the condition $\ell > 2/\epsilon$ is satisfied) to obtain

$$P^\ell(J) \leq 2P^{2\ell} \left\{ \mathbf{X}\mathbf{X}' : \exists f \in \mathcal{F}, \left| \left\{ \mathbf{x}'_j \in \mathbf{X}' : f(\mathbf{x}'_j) < \min_{i\in[\ell]} f(\mathbf{x}_i) - 2\gamma \right\} \right| > \epsilon\ell/2 \right\}.$$

The next stage of the proof is to consider a γ_k-cover U of $\mathcal{F}$ with respect to the pseudo-metric $l_\infty^{\mathbf{X}\mathbf{X}'}$, where the existence of

$$\gamma_k = \min\{\gamma : \mathcal{N}(\gamma, \mathcal{F}, 2\ell) \leq 2^k\}$$

is guaranteed by Lemma 10. Suppose that for some $f \in \mathcal{F}$,

$$\left| \left\{ \mathbf{x}'_j \in \mathbf{X}' : f(\mathbf{x}'_j) < \min_{i\in[\ell]} f(\mathbf{x}_i) - 2\gamma \right\} \right| > \epsilon\ell/2.$$

There exists $\hat{f} \in U$ with $\|f - \hat{f}\|_{l_\infty^{\mathbf{X}\mathbf{X}'}} \leq \gamma_k \leq \gamma$ and therefore $f(\mathbf{x}) < \min_{i\in[\ell]} f(\mathbf{x}_i) - 2\gamma$ implies $\hat{f}(\mathbf{x}) < \min_{i\in[\ell]} \hat{f}(\mathbf{x}_i)$. This implies that $\hat{f}$ satisfies

$$\left| \left\{ \mathbf{x}'_j \in \mathbf{X}' : \hat{f}(\mathbf{x}'_j) < \min_{i\in[\ell]} \hat{f}(\mathbf{x}_i) \right\} \right| > \epsilon\ell/2.$$

Hence, we can upper bound the probability $P^\ell(J)$ by

$$P^\ell(J) \leq 2P^{2\ell}\left\{\mathbf{XX}'\colon \exists \hat{f} \in B, \left|\left\{\mathbf{x}'_j \in \mathbf{X}' : \hat{f}(\mathbf{x}'_j) < \min_{i\in[\ell]} \hat{f}(\mathbf{x}_i)\right\}\right| > \epsilon\ell/2\right\}.$$

By definition of k, the set U has at most 2^k elements and thus applying the union bound (Vapnik, 1979, e.g.) over these possibilities gives

$$P^\ell(J) \leq 2 \cdot 2^k P^{2\ell}\left\{\mathbf{XX}'\colon \left|\left\{\mathbf{x}'_j \in \mathbf{X}' : \hat{f}(\mathbf{x}'_j) < \min_{i\in[\ell]} \hat{f}(\mathbf{x}_i)\right\}\right| > \epsilon\ell/2\right\}.$$

The final stage is to apply the permutation argument to bound the probability of the event on the right hand side by the fraction of swapping permutations which leave the event true for any fixed double sample $\mathbf{XX}'$. If we order the values $\hat{f}(\mathbf{x})$ for $\mathbf{x} \in \mathbf{XX}'$ the event remains true only if the $\lceil \epsilon\ell/2 \rceil$ points with smallest values remain on the right hand side. Hence, the fraction of permutations is bounded by $2^{-\lceil \epsilon\ell/2 \rceil}$ giving an overall bound of

$$\begin{aligned} & P^\ell(J) & \leq 2^{k+1-\lceil \epsilon\ell/2 \rceil} \\ \Rightarrow \quad & P^\ell(J) & \leq 2^{k+1-\epsilon\ell/2}. \end{aligned}$$

We complete the proof by setting the right hand side equal to $2\delta/\ell$ and solving for ϵ, ensuring that for a fixed value of k, with probability at least $1 - 2\delta/\ell$

$$P\left\{\mathbf{x} : f(\mathbf{x}) < \min_{i\in[\ell]} f(\mathbf{x}_i) - 2\gamma\right\} \leq \epsilon(\ell, k, \delta) = \tfrac{2}{\ell}(k + \log \tfrac{\ell}{\delta}).$$

But the result is trivial for $k > \ell/2$ and so applying the union bound over the integer possibilities $1 \leq k \leq \ell/2$ ensures that it fails for some k with probability at most δ as required. ∎

Though Theorem 12 provides the basis of our analysis it suffers from a weakness that a single perhaps unusual point may significantly reduce the minimum $\min_{i\in[\ell]} f(\mathbf{x}_i)$. We therefore now consider a technique which will enable us to ignore some of the smallest values in the set $\{f(\mathbf{x}_i)\colon \mathbf{x}_i \in \mathbf{X}\}$ at a corresponding cost to the complexity of the function class. The technique was originally considered for analysing soft margin classification in (Shawe-Taylor and Cristianini, 1999) but we will adapt it for the unsupervised problem we are considering here.

We will remove minimal points by increasing their output values. This corresponds to having a non-zero slack variable ξ_i in the algorithm, where we use the class of linear functions in feature space in the application of the theorem. There are well-known bounds for the log covering numbers of this class. We measure the increase in terms of $\mathcal{D}$ (Definition 6).

Let $\mathcal{X}$ be an inner product space. The following definition introduces a new inner product space based on $\mathcal{X}$.

Definition 13 *Let $L(\mathcal{X})$ be the set of real valued non-negative functions f on $\mathcal{X}$ with support $\mathrm{supp}(f)$ countable, that is functions in $L(\mathcal{X})$ are non-zero for at most countably many points. We define the inner product of two functions $f, g \in L(\mathcal{X})$, by*

$$f \cdot g = \sum_{\mathbf{x}\in \mathrm{supp}(f)} f(\mathbf{x})g(\mathbf{x}).$$

The 1-norm on $L(\mathcal{X})$ is defined by $\|f\|_1 = \sum_{\mathbf{x}\in\mathrm{supp}(f)} f(\mathbf{x})$ and let $L^B(\mathcal{X}) := \{f \in L(\mathcal{X}): \|f\|_1 \leq B\}$. Now we define an embedding of $\mathcal{X}$ into the inner product space $\mathcal{X} \times L(\mathcal{X})$ as follows.

$$\begin{aligned} \tau: \mathcal{X} &\rightarrow \mathcal{X} \times L(\mathcal{X}) \\ \tau: \mathbf{x} &\mapsto (\mathbf{x}, \delta_x), \end{aligned}$$

where $\delta_x \in L(\mathcal{X})$ is defined by

$$\delta_x(\mathbf{y}) = \begin{cases} 1, & \text{if } \mathbf{y} = \mathbf{x}; \\ 0, & \text{otherwise.} \end{cases}$$

For a function $f \in \mathcal{F}$, a set of training examples $\mathbf{X}$, and $\theta \in \mathbb{R}$, we define the function $g_f \in L(\mathcal{X})$ by

$$g_f(\mathbf{y}) = g_f^{\mathbf{X},\theta}(\mathbf{y}) = \sum_{\mathbf{x}\in\mathbf{X}} d(\mathbf{x}, f, \theta)\delta_x(\mathbf{y}).$$

Theorem 14 *Fix $B > 0$. Consider a fixed but unknown probability distribution P which has no atomic components on the input space $\mathcal{X}$ and a sturdy class of real valued functions $\mathcal{F}$. Then with probability $1 - \delta$ over randomly drawn training sequences $\mathbf{X}$ of size ℓ, for all $\gamma > 0$ and any $f \in \mathcal{F}$ and any θ such that $g_f = g_f^{\mathbf{X},\theta} \in L^B(\mathcal{X})$, (i.e. $\sum_{\mathbf{x}\in\mathbf{X}} d(\mathbf{x}, f, \theta) \leq B$.)*

$$P\{\mathbf{x}: f(\mathbf{x}) < \theta - 2\gamma\} \leq \tfrac{2}{\ell}(k + \log\tfrac{\ell}{\delta}), \tag{36}$$

where $k = \left\lceil \log \mathcal{N}(\gamma/2, \mathcal{F}, 2\ell) + \log \mathcal{N}(\gamma/2, L^B(\mathcal{X}), 2\ell) \right\rceil$.

(The assumption on P can in fact be weakened to require only that there is no point $\mathbf{x} \in \mathbf{X}$ satisfying $f(\mathbf{x}) < \theta - 2\gamma$ that has discrete probability.)

Proof We first form the space $\mathcal{X} \times L(\mathcal{X})$ and embed the space $\mathcal{X}$ into it with the mapping τ of Definition 13. We extend the class of functions $\mathcal{F}$ acting on $\mathcal{X}$ to the class $\mathcal{F} + L(\mathcal{X}) := \{f + g: f \in \mathcal{F}, g \in L(\mathcal{X})\}$, whose action is defined by

$$\begin{aligned} (f+g): \mathcal{X} \times L(\mathcal{X}) &\rightarrow \mathbb{R} \\ (f+g): (\mathbf{x}, h) &\mapsto f(\mathbf{x}) + g \cdot h. \end{aligned}$$

Note that the definition of sturdiness of a class of functions $\mathcal{F}$ only requires $S_{\mathbf{X}}(\mathcal{F})$ be compact. We can thus conclude that $S_{\mathbf{X}}(L^B(\mathcal{X}))$ is compact for any $\mathbf{X} \in \mathcal{X}$. Since $S_{\mathbf{X}}(\mathcal{F} + L^B(\mathcal{X})) = S_{\mathbf{X}}(\mathcal{F}) + S_{\mathbf{X}}(L^B(\mathcal{X}))$ and $\mathcal{F}$ is sturdy by assumption, we conclude that $\mathcal{F} + L^B(\mathcal{X})$ is also sturdy.

We can bound the log covering numbers of the space $\mathcal{F} + L^B(\mathcal{X})$ in terms of the component spaces as follows. For any $\gamma > 0$,

$$\log \mathcal{N}(\gamma, \mathcal{F} + L^B(\mathcal{X}), \ell) \leq \log \mathcal{N}(\gamma/2, \mathcal{F}, \ell) + \log \mathcal{N}(\gamma/2, L^B(\mathcal{X}), \ell),$$

since we can create a cover for $\mathcal{F} + L^B(\mathcal{X})$ from pairs of functions taken from the two component covers. We can therefore apply Theorem 12 to the combined function class. The function $(f + g_f) \in \mathcal{F} + L^B(\mathcal{X})$ satisfies the following two properties:

1. For $\mathbf{x} \in \mathbf{X}$ we have

$$(f + g_f)(\tau(\mathbf{x})) = f(\mathbf{x}) + d(\mathbf{x}, f, \theta) \geq \theta,$$

and therefore $\min_{i \in [\ell]}(f + g_f)(\tau(\mathbf{x}_i)) \geq \theta$.

2. For $\mathbf{x} \notin \mathbf{X}$,

$$(f + g_f)(\tau(\mathbf{x})) = f(\mathbf{x}).$$

Denote by J the event in (36). Observe that $\mathbf{x} \notin \mathbf{X} \Rightarrow (f(\mathbf{x}) < \theta - 2\gamma \Rightarrow (f+g_f)(\tau(\mathbf{x})) < \theta - 2\gamma)$. Thus up to a set of measure zero, $J \subseteq \{\mathbf{x}\colon (f+g_f)(\tau(\mathbf{x})) < \theta - 2\gamma\} =: K$. But Theorem 12 implies $P(K) \leq \frac{2}{\ell}(k + \log \frac{\ell}{\delta})$. ∎

The theorem bounds the probability of a new example falling in the region for which $f(\mathbf{x})$ has value less than $\theta - 2\gamma$, this being the complement of the estimate for the support of the distribution. In the algorithm described in this paper, one would use the hyperplane shifted by 2γ towards the origin to define the region. Note that there is no restriction placed on the class of functions though these functions could be probability density functions.

The result shows that we can bound the probability of points falling outside the region of estimated support by a quantity involving the ratio of the log covering numbers (which can be bounded by the fat shattering dimension at scale proportional to γ) and the number of training examples, plus a factor involving the 1-norm of the slack variables. The result is stronger than related results given by Ben-David and Lindenbaum (1997): their bound involves the square root of the ratio of the Pollard dimension (the fat shattering dimension when γ tends to 0) and the number of training examples.

We now derive a result from the theorem for the specific situation considered in this paper, namely where the choice of function class is the class of linear functions in a kernel defined feature space. The following lemma follows immediately from Lemma 7.14 of (Shawe-Taylor and Cristianini, 2000).

Lemma 15 *For all $\gamma > 0$,*

$$\log \mathcal{N}(\gamma, L^B(\mathcal{X}), \ell) \leq b \log \left(\frac{e(\ell + b - 1)}{b} \right),$$

where $b = \left\lfloor \frac{B}{2\gamma} \right\rfloor$.

We use the following covering number bound for the linear function class itself.

Theorem 16 (Williamson et al. (2000)) *Let $\mathcal{F}$ be the class of linear classifiers with norm at most 1 confined to a ball of radius 1 about the origin, then for $\epsilon \geq c/\sqrt{\ell}$,*

$$\log \mathcal{N}(\epsilon, \mathcal{F}, \ell) \leq \frac{c^2 \log(\frac{2 \ln(2)}{c^2} \epsilon^2 \ell)}{\epsilon^2}$$

where $c = 103$.

(If $\epsilon \leq c/\sqrt{\ell}$ the bound is $\geq \ell$ and so the application of it below would be trivial in any case. Note too that by setting $\epsilon' = \epsilon/R$ the result extends to functions confined to any radius $R > 0$ trivially.)

We are now ready to present the theorem that bounds the probabilities of falling outside the estimated region of support for the algorithm described in the main body of the paper. The functions considered are linear functions in the kernel defined feature space. The estimation depends on the norm of the function f_w, which can be determined from the parameters by

$$\|w\|^2 = \sum_{i,j \in [\ell]} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j).$$

Theorem 17 *Consider a fixed but unknown probability distribution P with no atomic components on the space $\mathcal{X}$ with support contained in a ball of radius 1 about the origin and a linear class of functions on $\mathcal{X}$ defined by $\mathcal{F} := \{f_w \colon \mathbf{x} \mapsto \mathbf{x} \cdot w \in \mathbb{R} \colon\ w \in \mathbb{R}^{\dim \mathcal{X}}\}$. Then with probability $1 - \delta$ over randomly drawn training sequences $\mathbf{X}$ of size ℓ, for all $\gamma > 0$ and any $f_w \in \mathcal{F}$,*

$$P\{\mathbf{x} \colon f_w(\mathbf{x}) < \theta - 2\gamma\} \leq \frac{2}{\ell}\left(k + \log\frac{\ell^2}{2\delta}\right),$$

where

$$k = \frac{c_1 \log(c_2 \hat{\gamma}^2 \ell)}{\hat{\gamma}^2} + \frac{\mathcal{D}}{\hat{\gamma}} \log\left(e\left(\frac{(2\ell - 1)\hat{\gamma}}{\mathcal{D}} + 1\right)\right) + 2,$$

$c_1 = 4c^2$, $c_2 = \ln(2)/c^2$, $c = 103$, $\hat{\gamma} = \gamma/\|w\|$, *and* $\mathcal{D} = \mathcal{D}(\mathbf{X}, f_w, \theta)$.

Proof First observe that by applying a fixed shift to all the functions we can without loss of generality treat the threshold as 0. We wish to apply Theorem 14 to the linear function class $\mathcal{F}$ with norm no greater than 1. Hence, we rescale the function to obtain $\hat{f} = f_{w/\|w\|}$ and $\hat{g} = g_{f_w/\|w\|} = g_{f_w}/\|w\|$. Note that the function f_w and margin γ define the same region as $\hat{f}$ with margin $\hat{\gamma} = \gamma/\|w\|$. Furthermore,

$$\mathcal{D}(\mathbf{X}, \hat{f}, 0) = \mathcal{D}(\mathbf{X}, f_w, 0)/\|w\|.$$

We have to take account of the fact that the B of Theorem 14 is not fixed here, but is allowed to depend on the data sample. We thus stratify over possible values of B. We apply the theorem for each value of

$$\left\lceil \log \mathcal{N}(\hat{\gamma}/2, L^B(\mathcal{X}), 2\ell) \right\rceil,$$

where $B = \mathcal{D}(\mathbf{X}, \hat{f}, 0)$, so that it suffices to make at most $\ell/2$ applications for the possible non-trivial bounds (using a confidence of $\delta/(\ell/2)$ for each bound). Using Lemma 15 and Theorem 16 we obtain the following bound on the covering numbers. Let $b = \lfloor \frac{B}{\hat{\gamma}} \rfloor$. Then we have

$$\begin{aligned} \lceil \log \mathcal{N}(\hat{\gamma}/2, \mathcal{F}, 2\ell) \quad &+ \quad \left\lceil \log \mathcal{N}(\hat{\gamma}/2, L^B(\mathcal{X}), 2\ell) \right\rceil \rceil \\ &\leq \quad \left\lceil \frac{c^2 4 \log\left(\frac{\ln(2)}{c^2}\hat{\gamma}^2\ell\right)}{\hat{\gamma}^2} + \left\lceil b \log\left(\frac{e(2\ell + b - 1)}{b}\right)\right\rceil \right\rceil \end{aligned}$$

$$
\begin{aligned}
&\leq \frac{c_1 \log(c_2 \hat{\gamma}^2 \ell)}{\hat{\gamma}^2} + b \log\left(\frac{e(2\ell + b - 1)}{b}\right) + 2 \\
&\leq \frac{c_1 \log(c_2 \hat{\gamma}^2 \ell)}{\hat{\gamma}^2} + \frac{\mathcal{D}}{\hat{\gamma}} \log\left(\frac{e(2\ell + (\mathcal{D}/\hat{\gamma}) - 1)\hat{\gamma}}{\mathcal{D}}\right) + 2 \\
&= \frac{c_1 \log(c_2 \hat{\gamma}^2 \ell)}{\hat{\gamma}^2} + \frac{\mathcal{D}}{\hat{\gamma}} \log\left(e\left(\frac{(2\ell - 1)\hat{\gamma}}{\mathcal{D}} + 1\right)\right) + 2,
\end{aligned}
$$

where the last inequality follows from the fact that $b \log(e(2\ell - b + 1)/b)$ is monotonic in b. Note, moreover, that we have been somewhat sloppy in not excluding the case $b = 0$ in the denominator of the log argument. The argument, however, goes through if we take into account the b in front of the log and replace the whole term by its limit as b approaches 0. ∎

Now we can develop a version of the above theorem directly applicable to the algorithm presented in this paper.

Proof (Theorem 7) The proof is immediate upon identifying θ with ρ and rescaling 2γ to γ. ∎