Kernel Method for Percentile Feature Extraction

Bernhard Schölkopf*, John C. Platt‡, Alex J. Smola§

* Microsoft Research, 1 Guildhall Street, Cambridge CB2 3NH, UK

‡ Microsoft Research, 1 Microsoft Way, Redmond, WA, USA

§ Department of Engineering, Australian National University, Canberra 0200, Australia

bsc@microsoft.com, jplatt@microsoft.com, Alex.Smola@anu.edu.au

2 February 2000

Technical Report
MSR-TR-2000-22

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Abstract

A method is proposed which computes a direction in a dataset such that a specified fraction of a particular class of all examples is separated from the overall mean by a maximal margin. The projector onto that direction can be used for class-specific feature extraction. The algorithm is carried out in a feature space associated with a support vector kernel function, hence it can be used to construct a large class of nonlinear feature extractors. In the particular case where there exists only one class, the method can be thought of as a robust form of principal component analysis, where instead of variance we maximize percentile thresholds. Finally, we generalize it to also include the possibility of specifying negative examples.

1 Introduction and Notation

Suppose we are given two sets of data: a set of points

$$\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_t\} \tag{1}$$

which we think of as representative of the kind of data that we typically encounter in some problem of interest, and a second set

$$\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_\ell\} \tag{2}$$

representing a specific class of examples that we are interested in.

Our goal in the present paper is to construct a real-valued *feature extractor* which, given a previously unseen test point $\mathbf{x}$, charaterizes the "$\mathbf{X}$-ness" of the point $\mathbf{x}$. By this we mean a feature extractor which takes large values for points similar to those in $\mathbf{X}$ and small values for generic points from $\mathbf{Z}$. In addition to this, we will compute a *threshold* value such that if we draw a novel point from the same distribution $P(\mathbf{x})$ as the one underling $\mathbf{X}$, then its feature value will be above threshold with, approximately, a pre-specified probability. In this sense, we are trying to estimate regions that contain specified fractions of the probability mass of $P(\mathbf{x})$, including the case where we estimate the whole support of $P(\mathbf{x})$. At the same time, if a novel point falls below that threshold, then we can assert that it is unlikely to have been generated from $P(\mathbf{x})$. This task is referred to as novelty (or anomaly) detection. It has ample applications, but there are few approaches that are viable for high-dimensional data. Moreover, compared to widely studied problems such as pattern recognition or density estimating, there exists little theory dealing with novelty detection (cf. (Ben-David & Lindenbaum, 1997)). This stands in sharp contrast to the practical importance that novelty detection tasks have assumed, for instance in medical diagnosis (Tarassenko et al., 1995).

The present approach can be used for the estimation of a distribution's support, for novelty detection, and for feature extraction. It employs two ideas from support vector machines (Vapnik, 1995) which we are crucial for their fine

generalization performance even in high-dimensional tasks. Those are the idea of maximizing a margin, and the idea of nonlinearly mapping the data into some *feature space* F. In the remainder of this section, we shall describe the latter.

We assume that the feature space be endowed with a dot product. This need not be the case for the input domain $\mathcal{X}$; in fact, we do not even require that it be a vector space; for instance, it could be a discrete set. The connection between the input domain and the feature space is established by a feature map

$$\Phi : \mathcal{X} \to F, \tag{3}$$

i.e. a map such that some simple kernel (Boser et al., 1992; Vapnik, 1995)

$$k(\mathbf{x}, \mathbf{y}) = (\Phi(\mathbf{x}) \cdot \Phi(\mathbf{y})), \tag{4}$$

such as the Gaussian

$$k(\mathbf{x}, \mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2/c}, \tag{5}$$

provides a dot product in the image of Φ. In practice, we need not necessarily worry about Φ, as long as a given k satisfies certain positivity conditions (Vapnik, 1995).

As F is a dot product space, we can use tools of linear algebra and geometry to construct algorithms in F, even if the input domain $\mathcal{X}$ is discrete. Below, we derive our results in F, using the following shorthands:

$$x_i = \Phi(\mathbf{x}_i) \tag{6}$$

$$z_n = \Phi(\mathbf{z}_n) \tag{7}$$

$$X = \{x_1, \ldots, x_\ell\} \tag{8}$$

$$Z = \{z_1, \ldots, z_t\} \tag{9}$$

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

2 Algorithms

In analogy to an algorithm recently proposed for the estimation of a distribution's support (Schölkopf et al., 1999), we will try to construct a nonlinear decision function on $\mathcal{X}$ by mapping the data into some feature space and then seeking to separate X from the centroid of Z with a large margin hyperplane

committing few training errors. Projections on the normal vector of the hyperplane then characterize the "X-ness" of test points, and the area where the decision function takes the value 1 can serve as an approximation of the support of X. While X is the set that we are actually interested in, the set Z thus only plays the role of, in some weak and possible imprecise sense, modeling what the unknown "other" examples might look like. This will be useful if our algorithm for estimating a distribution's support is applied to problems such as novelty detection.

In analogy to ν-support-vector algorithms (Schölkopf et al., 2000), the decision function is found by minimizing a weighted sum of a support vector type regularizer and an empirical error term depending on an overall margin variable ρ and individual L_1 errors ξ_i,

$$\min_{w \in F, \xi \in \mathbb{R}^\ell, \rho \in \mathbb{R}} \quad \tfrac{1}{2}\|w\|^2 + \tfrac{1}{\nu\ell}\textstyle\sum_i \xi_i - \rho \tag{10}$$

$$\text{subject to} \quad (w \cdot (x_i - \tfrac{1}{t}\textstyle\sum_n z_n)) \geq \rho - \xi_i \tag{11}$$

$$\xi_i \geq 0. \tag{12}$$

The precise meaning of the parameter ν governing the trade-off between the regularizer and the traning error 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

$$f(x) = \operatorname{sgn}((w \cdot (x - \frac{1}{t}\sum_n z_n)) - \rho) \tag{13}$$

will be positive for many examples x_i contained in X, while the SV type regularization term $\|w\|$ will still be small. This can be shown to correspond to a large margin of separation from $\frac{1}{t}\sum_n z_n$.

We next compute a dual form of this optimization problem. To this end, we introduce Lagrange multipliers $\alpha_i, \beta_i \geq 0$, and a Lagrangian

$$L(w, \xi, \rho, \alpha, \beta) = \frac{1}{2}\|w\|^2 + \frac{1}{\nu\ell}\sum_i \xi_i - \rho$$

$$- \sum_i \alpha_i((w \cdot (x_i - \frac{1}{t}\sum_n z_n)) - \rho + \xi_i) - \sum_i \beta_i \xi_i. \tag{14}$$

The Lagrangian needs to be maximized with respect to the primal variables w, ξ, ρ, and minimized with respect to the Lagrange multipliers. We first set the derivatives with respect to the primal variables equal to zero, yielding

$$w = \sum_i \alpha_i (x_i - \frac{1}{t}\sum_n z_n), \tag{15}$$

$$\alpha_i = \frac{1}{\nu\ell} - \beta_i \leq \frac{1}{\nu\ell}, \tag{16}$$

$$\sum_i \alpha_i = 1. \tag{17}$$

In (15), all patterns $\{\mathbf{x}_i : i \in [\ell],\ \alpha_i > 0\}$ are called Support Vectors. The expansion (15) turns the decision function (13) into a form which only depends on dot prducts,

$$f(x) = \mathrm{sgn}((\sum_i \alpha_i (x_i - \frac{1}{t}\sum_n z_n) \cdot (x - \frac{1}{t}\sum_n z_n)) - \rho). \tag{18}$$

By multiplying out the dot products, we obtain a form that can be written as a nonlinear decision function on the input domain $\mathcal{X}$ in terms of a kernel (4) (cf. (6) and (7))

$$\begin{aligned} f(\mathbf{x}) &= \mathrm{sgn}\Big(\sum_i \alpha_i \big(k(\mathbf{x}_i, \mathbf{x}) - \frac{1}{t}\sum_n k(\mathbf{z}_n, \mathbf{x}) \\ &\quad + \frac{1}{t^2}\sum_{np} k(\mathbf{z}_n, \mathbf{z}_p) - \frac{1}{t}\sum_n k(\mathbf{z}_n, \mathbf{x}_i)\big) - \rho\Big) \qquad (19) \\ &= \mathrm{sgn}\Big(\sum_i \alpha_i k(\mathbf{x}_i, \mathbf{x}) - \frac{1}{t}\sum_n k(\mathbf{z}_n, \mathbf{x}) \\ &\quad + \frac{1}{t^2}\sum_{np} k(\mathbf{z}_n, \mathbf{z}_p) - \frac{1}{t}\sum_{in} \alpha_i k(\mathbf{z}_n, \mathbf{x}_i) - \rho\Big) \qquad (20) \end{aligned}$$

Note that the last step used the constraint (17), and that we have slightly abused the symbol f by employing it to denote both the decision function in feature space (18) and in input space (20). Moreover, in the argument of the sgn, only the first two terms depend on $\mathbf{x}$, therefore we may absorb the next terms in the constant ρ, which we have not fixed yet. To compute ρ in the final form of the decision function

$$f(\mathbf{x}) = \mathrm{sgn}\left(\sum_i \alpha_i k(\mathbf{x}_i, \mathbf{x}) - \frac{1}{t}\sum_n k(\mathbf{z}_n, \mathbf{x}) - \rho\right), \tag{21}$$

we employ the KKT conditions of the optimization problem (Bertsekas, 1995, e.g.). They state that for points $\mathbf{x}_i$ where both α_i and β_i are nonzero, the inequality constraints (11) and (12) become equalities. In other words, if $\alpha_i \in (0, 1/(\nu\ell))$ (note that in general, $\alpha_i \in [0, 1/(\nu\ell)]$), then the argument of the sgn in the decision function should equal 0, i.e. the corresponding $\mathbf{x}_i$ sits exactly on the hyperplane of separation.

The KKT conditions also imply that only those points $\mathbf{x}_i$ can have a nonzero α_i for which the inequality constraint in (11) is precisely met; therefore the support vectors ξ with $\alpha_i > 0$ will often form but a small subset of $\mathbf{X}$. However, the solution depends on all $\mathbf{z}_n$, hence it will not necessarily be particularly sparse. If this is a concern, then postprocessing can be applied to increase sparsity, along the lines of Schölkopf et al., 1999.

Substituting (15) – (17) into L (14), we can eliminate the primal variables to get the dual problem. A short calculation shows that it consists of minimizing

the quadratic form

$$\begin{aligned} W(\alpha) &= \frac{1}{2}\sum_{ij}\alpha_i\alpha_j\left((x_i\cdot x_j)+q-q_j-q_i\right) \\ &= \frac{1}{2}\sum_{ij}\alpha_i\alpha_j\left(k(\mathbf{x}_i,\mathbf{x}_j)+q-q_j-q_i\right), \end{aligned} \tag{22}$$

using the shorthands $q := \frac{1}{t^2}\sum_{np}k(\mathbf{z}_n,\mathbf{z}_p)$ and $q_j := \frac{1}{t}\sum_n k(\mathbf{x}_j,\mathbf{z}_n)$, subject to the constraints

$$0 \le \alpha_i \le \frac{1}{\nu\ell}, \quad \sum_i \alpha_i = 1. \tag{23}$$

This convex quadratic program can be solved with standard quadratic programming tools. Alternatively, one can employ the SMO algorithm described in (Schölkopf et al., 1999), which was found to approximately scale quadratically with the training set size.

Finally, it is interesting to note that kernel PCA, which is normally formulated as an eigenvalue problem, solves almost the same problem. The target function is the same, however it is minimized subject to the constraint that the variance of $f(\mathbf{x}) = \sum_i \alpha_i k(\mathbf{x}_i,\mathbf{x})$ on the training set be 1.

3 Determining Percentiles Using ν

Note that if ν approaches 0, the upper boundaries on the Lagrange multipliers tend to infinity, i.e. the second inequality constraint in (22) 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 (10). The dual problem is still feasible, since we have placed no restriction on ρ, so ρ can become a large negative number in order to satisfy (11). If we had required $\rho \ge 0$ from the start, we would have ended up with the constraint $\sum_i \alpha_i \ge 1$ instead of the corresponding equality constraint in (22), and the multipliers α_i could have diverged.

Next, assume that $\nu = 1$. In this case, the constraints (23) alone already determine the solution: all α_i must equal $1/\ell$, the argument of the decision function (21) then reduces to the difference between two *Parzen windows* density estimates, one for $\mathbf{X}$ and one for $\mathbf{Z}$, and the decision function is a thresholded version of that difference,

$$f(\mathbf{x}) = \operatorname{sgn}\left(\frac{1}{\ell}\sum_i k(\mathbf{x}_i,\mathbf{x}) - \frac{1}{t}\sum_n k(\mathbf{z}_n,\mathbf{x}) - \rho\right). \tag{24}$$

This amounts to a thresholded variant of the Bayes decision boundary for the considered density models. Note that in a Parzen windows estimator, the kernels are required to have integral 1, i.e. to be densities of probability measures.

However, since we are thresholding them anyway, a global rescaling factor can be disregarded.

Note, moreover, that the constraints (23) rule out solutions where $\nu > 1$, as in that case, the α_i cannot sum up to 1. Negative values of ν are ruled out, too, since they would amount to encouraging (rather than penalizing) training errors in (10). Therefore, in the primal problem (10) only $\nu \in (0, 1]$ makes sense. We shall now explain that ν actually characterizes how many points of $\mathbf{X}$ are allowed to lie outside the region where the decision function is positive. To this end, we introduce the term *outlier* to denote points $\mathbf{x}_i$ that have a nonzero slack variable ξ_i, i.e. points that lie outside of the estimated region. By the KKT conditions, all outliers are also support vectors; however there can be support vectors (sitting exactly on the margin) that are not outliers.

Proposition 1 *Assume the solution of (10) 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 (8) were generated independently from a distribution $P(x)$ 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.

Parts (i) and (ii) can be proven directly based on the primal objective function (10), as sketched presently: suppose we have found the solution. If we now decrease ρ, the term $\sum_i \xi_i$ will change proportionally to the *number* of points that have a nonzero ξ_i (the outliers). If we *increase* ρ, it will be proportional to the number of points which are either already outliers, or just about to get a nonzero ρ, i.e. which sit *on* the hyperplane — taken together, the set of SVs. At the optimum of (10), 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 (Schölkopf et al., 2000).

From the proof, note that the statements (i) and (ii) are precise in the sense that if we changed the ρ that the algorithm comes up with by some $\epsilon > 0$, then one of the two would not hold anymore. The statements do not assert, however, that the proposed algorithm maximizes the margin *subject to* the constraint that only a fraction ν of outliers is allowed. This problem would be a combinatorial one; using a convex program, such as in our algorithm, its solution can therefore only be approximated.

Our experience suggests that the approximation obtained is a good one, however it is an open theoretical question whether a precise statement to that effect can be made.

4 Special Cases and Extensions

Separation from the origin. Assume that there is only one z_n, equalling 0, i.e. we are trying to separate the data from the origin in F. In this case, both the decision function and the optimization problem reduce to what is described in (Schölkopf et al., 1999). Note that the connection to the Parzen windows density estimator noted in Sec. 3 also applies in the present case. Here, it states that for $\nu = 1$, the decision function will be nothing but a thresholded version of a Parzen density estimate. As ν gets smaller than 1, fewer points will appear in the expansion (cf. Proposition 1). Therefore, it will behave like a thresholded version of a Parzen density estimator where some kernels have been pruned. The pruning is such that the most typical examples are thrown out first — remember that the SVs are either outliers or on the edge of the decision boundary. This makes perfect sense, as for the task of estimating the support of a distribution, rather then its density, it is irrelevant to represent the density inside the estimated area — only the boundaries count.

Separating a dataset from its mean. To give meaning to this somewhat paradoxical phrase, let us start by assuming that $\mathbf{X} = \mathbf{Z}$. In this case, we are separating the data points from their mean. Note that we have not ruled out the case of a negative ρ, so this will be feasible. As in (Schölkopf et al., 1999), it can be shown that the margin of separation to the centroid is $\rho/|w|$, hence the margin can be negative, too. At the same time, some of the training examples will usually lie outside of the estimated region; the number of such examples is controlled by the trade-off constant ν (Proposition 1). Suppose, for instance, we adjust $\nu = 90\%$, such that 10% of all examples are in the estimated region. Then the solution effectively gives us a direction and an offset such that along that direction, 10% of the data are further away from the mean than that offset, and the offset is as large as possible (cf. the notes following Proposition 1). It is thus giving us feature extractors with large percentile offsets. Principal component analysis, in contrast, provides feature extractors with large variance, either directly in input space, or, in the case of kernel PCA (Schölkopf et al., 1998), in feature space. Using variance as a contrast function, however, is known to be sensitive towards outliers, whereas the present algorithm can be shown to have a desirable resistance property (see Sec. 5).

So far, we have only told you how to compute the first feature extraction direction. Higher order features can be extracted subsequently by projecting out the previous direction from the dataset. For instance, if w is the first direction, we generate a new dataset by transforming the points according to

$$x_i' := x_i - \left(\frac{w}{|w|} \cdot x_i\right) \frac{w}{|w|}, \tag{25}$$

$$z_n' := z_n - \left(\frac{w}{|w|} \cdot z_n\right) \frac{w}{|w|}. \tag{26}$$

Note that this ensures that $(w \cdot x_i') = (w \cdot z_n') = 0$, therefore all the transformed

points are in the subspace orthogonal to w.

Separation of a subset from the overall mean. Here, we consider the situation where $\mathbf{X} \subset \mathbf{Z}$. This is a generalization of the previous case, where the feature extractor is only seeking to characterize a subset of the whole dataset.

Incorporating negative examples. Suppose now that we are given information in addition to $\{x_1, \ldots, x_\ell\}$, namely, a set of corresponding labels $\{y_1, \ldots, y_\ell\}$, where $y_i \in \{\pm 1\}$. As in the theoretical results of (Schölkopf et al., 1999), we point-reflect those x_i that have a negative y_i, and then solve the resulting problem as above. The point reflection is carried out w.r.t. the mean of Z. In the primal problem, this leads to a y_i on the left hand side of the constraint (11). In the support vector expansion (15), α_i is replaced by $\alpha_i y_i$, the same applies to the dual objective function (22) and to the decision function. Note, however, that we have to use the decision function in the form (19), not (20), as the last simplification does not hold in this more general case. Everything else remains the same. Note that our original algorithm is contained as a special case, with all labels equal to $+1$. This approach applies no matter whether the sets of positive and negative labels are balanced or not.

5 Theoretical Results

A number of results proven for the special case of separation from the origin carry over to the more general case. It would be redundant to reproduce them in the present work, we refer to (Schölkopf et al., 2000) for details. Among them are a statement about the optimality (in the sense of maximizing the margin) of the computed hyperplane in the case where X is linearly separable from the mean of Z, and some insights characterizing the connection to binary classification. Moreover, the above paper contains a proof of a resistance results that we briefly re-state here:

Proposition 2 (Resistance) *Local movements of outliers parallel to* $\mathbf{w}$ *do not change the hyperplane.*

We refrain from reproducing the proof. Essentially, the result is due to the fact that the errors ξ_i enter in the objective function only linearly. To determine the hyperplane, we need to find the (constrained) extremum of the objective function, and in finding the extremum, the derivatives are what counts. For the linear error term, however, those are constant, so they do not depend on how far away from the hyperplane an error point lies.

Finally, the paper dealing with separation from the origin contains generalization error bounds. Roughly, they state the following: suppose the estimated hyperplane separates part of X from the origin by a certain margin, and with an offset ρ. Now we are given test examples coming from the same distribution as X, and classify them using a shifted hyperplane, with offset $\eta\rho$, where $\eta < 1$.

Then the probability that they are on the wrong side of the shifted hyperplane will exceed the fraction of training outliers at most by a complexity term that depends on η (the further η from 1, the smaller it is; for $\eta \to 1$, it diverges) and on the margin (the larger the margin, the smaller the term).

Note that $\eta < 1$ means that the region is somewhat larger than the one determined by the algorithm. As η approaches 1, the region that the generalization error bound talks about approaches the region returned by the algorithm; at the same time, the bound gets weaker.

Note that originally, the bound refers to the separation from the origin. We expect that the extension to separation from points estimated from the data will not cost much in terms of increasing the complexity term. Indeed, if X and Z are disjoint, then the bound does not change at all, since it that case to compute the mean of Z we need not use any data from X.

6 Experiments

In the section, we show some preliminary experiments. Figure 1 shows a toy example of an estimation of a region that separates a class of data from the mean of another set. Note that the latter mean is taken in feature space, indeed, if we were to separate the circles from the mean of the asterisks in input space, the decision boundary would look different. As the mean in feature space, $\frac{1}{t}\sum_n z_n$, does not normally correspond to a single point in input space, it (in a sense) retains information about all the asterisks. This is, to some extent, noticable from the shape of the decision boundary, which takes into accound the asterisks to a much finer degree than just looking at their mean.

Next, we show some results on real-world data, obtained for the special case of separation from the mean (from (Schölkopf et al., 1999)). We used 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 (5) of width $c = 0.5 \cdot 256$, a common value for SVM classifiers on that data set, 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. 2) contains a number of patterns which are hard or impossible to classify, due to segmentation errors or mislabelling (Vapnik, 1995, e.g.). In the 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. Vice versa, with the labels, the algorithm has the chance to identify both unusual patterns and usual patterns with unusual labels. Fig. 3 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. As in the toy example, we used a ν value of 5%.

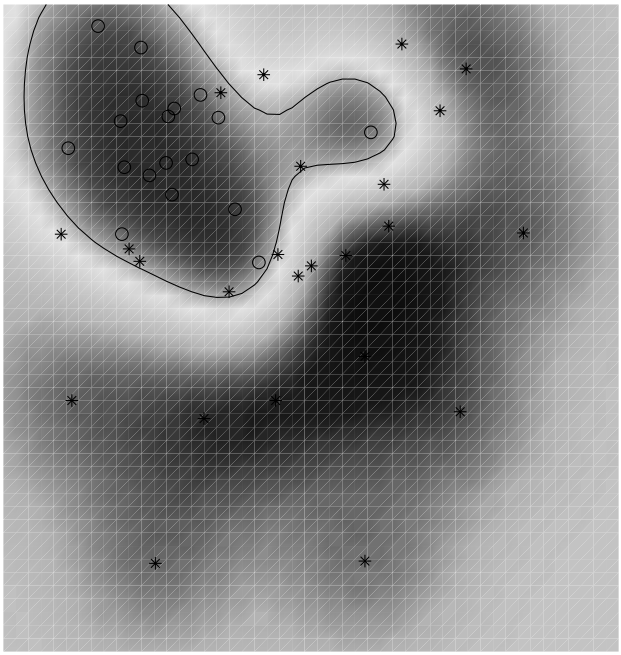

Figure 1: Toy example of the proposed algorithm separating one class (marked by circles) from the mean, taken in feature space, of a collection of "generic" examples (marked by aterisks). In the experiment, we used $\nu = 5\%$ and the Gaussian kernel (5) with $c = 0.1$. As threshold, we used $0.9 \cdot \rho$, where ρ was the one returned by the algorithm. As argued in Sec. 5, this is preferable to using ρ from a theoretical point of view. In the picture, it slightly enlarges the decision region — otherwise, several of the circles which are inside the estimated region would exactly sit on the decision boundary (the SVs).

7 Discussion

The present work builds on our previous algorithm for estimating a distribution's support. That algorithm, separating the data from the origin in feature space, suffered from the drawback that the origin played a special role. One way to think of it is as a prior on where, in a novelty detection context, the unknown "other" class lies. The present work alleviates this problem by allowing for the possibility to separate from a point inferred from the data, either from the same class, or from some other data. Serendipitously, this has led to robust PCA type algorithms in feature space computing feature extractors maximizing percentile thresholds.

There is a concern that one could put forward about one of the variants of the presently proposed approach, namely about the case where X and Z are disjoint, and we are separating X from Z's centroid: why not actually train a

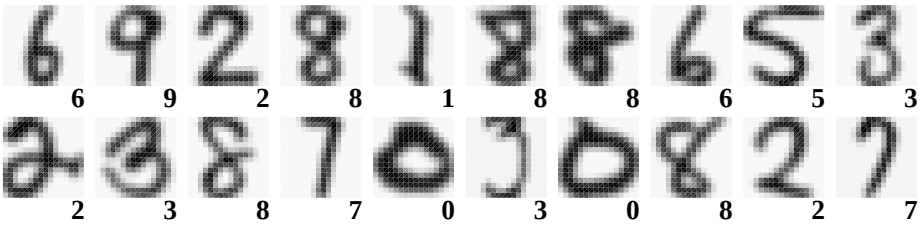

Figure 2: A subset of 20 examples randomly drawn from the USPS test set, with class labels.

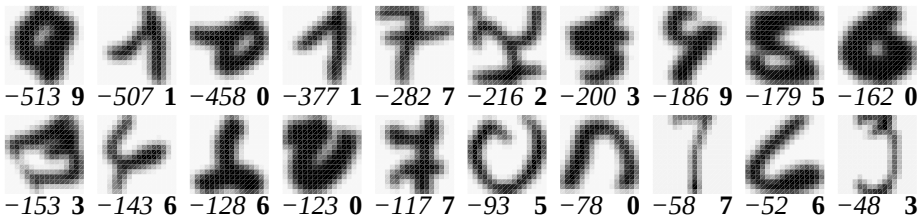

Figure 3: Outliers identified by the proposed algorithm, ranked by the negative output of the SVM (the argument of (21)). The outputs (for convenience in units of 10^{-5}) are written underneath each image in italics, the (alleged) class labels are given in bold face. Note that most of the examples are "difficult" in that they are either atypical or even mislabelled.

full binary classifier separating X from all examples from Z, rather that just from its mean? Indeed there might be situations where this is appropriate. More specifically, whenever Z is representative of the instances of the other class that we expect to see in the future, then a binary classification is certainly preferable. However, there can be situations where Z is not representative for the other class, for instance due to nonstationarity. Maybe it is even the case that Z only consists of artificial examples. In this situation, the only real training examples are the positive ones. In this case, separating the data from the mean of some artificial, or non-representative examples, provides a way of taking into account *some* information from the other class which might work better than simply separating the positive data from the origin. However, this conjecture has yet to be confirmed in real-world experiments.

In any event, the present algorithm enlarges the toolkit for modelling distributions using support vector methods. These methods have a large number of applications that are currently being investigated, including tasks such as novelty detection.

The philosophy behind our approach is the one advocated by (Vapnik, 1995). If you are trying to solve a learning problem, do it directly, rather than solving a more general problem along the way. Applied to the estimation of a distribu-

tion's support, this means: do not first estimate a density and then threshold it to get an estimate of the support. In the present paper, we have shown that our direct approach contains a Parzen windows density estimation approach asa special case, however, it was the special case for $\nu = 1$, which is not the most sensible parameter choice conceivable. General values of ν, the outlier constant, provide a useful means of taking into account information on how much noise we expect in the data, and lead to thresholded kernel expansions which behave different from generic density estimates.

Acknowledgements

Thanks to Chris Bishop, Andrew Blake, Paul Hayton, John Shawe-Taylor, Lionel Tarassenko, Mike Tipping, and Bob Williamson for useful discussions.

References

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

Bertsekas, D. P. (1995). *Nonlinear programming.* Belmont, MA: Athena Scientific.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory* (pp. 144–152). Pittsburgh, PA: ACM Press.

Schölkopf, B., Mika, S., Burges, C., Knirsch, P., Müller, K.-R., Rätsch, G., & Smola, A. (1999). Input space vs. feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, *10*, 1000 – 1017.

Schölkopf, B., Platt, J., Shawe-Taylor, J., Smola, A., & Williamson, R. (1999). *Estimating the support of a high-dimensional distribution*TR MSR 99 - 87). Microsoft Research, Redmond, WA. Submitted to *Neural Computation.*

Schölkopf, B., Smola, A., & Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, *10*, 1299–1319.

Schölkopf, B., Smola, A., Williamson, R. C., & Bartlett, P. L. (2000). New support vector algorithms. *Neural Computation*, *12*, 1083 – 1121.

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

Vapnik, V. (1995). *The nature of statistical learning theory.* N.Y.: Springer.