Machine Learning, 54, 45–66, 2004

Support Vector Data Description

DAVID M.J. TAX davidt@first.fhg.de
ROBERT P.W. DUIN r.p.w.duin@tnw.tudelft.nl
Pattern Recognition Group, Faculty of Applied Sciences, Delft University of Technology, Lorentzweg 1, 2628 CJ Delft, The Netherlands

Editor: Douglas Fisher

Abstract. Data domain description concerns the characterization of a data set. A good description covers all target data but includes no superfluous space. The boundary of a dataset can be used to detect novel data or outliers. We will present the Support Vector Data Description (SVDD) which is inspired by the Support Vector Classifier. It obtains a spherically shaped boundary around a dataset and analogous to the Support Vector Classifier it can be made flexible by using other kernel functions. The method is made robust against outliers in the training set and is capable of tightening the description by using negative examples. We show characteristics of the Support Vector Data Descriptions using artificial and real data.

Keywords: outlier detection, novelty detection, one-class classification, support vector classifier, support vector data description

1. Introduction

Much effort has been expended to solve classification and regression tasks. In these problems a mapping between objects represented by a feature vector and outputs (a class label or real valued outputs) is inferred on the basis of a set of training examples. In practice, another type of problem is of interest too: the problem of data description or One-Class Classification (Moya, Koch, & Hostetler, 1993). Here the problem is to make a *description* of a training set of objects and to detect which (new) objects resemble this training set.

Obviously data description can be used for outlier detection—to detect uncharacteristic objects from a data set. Often these outliers in the data show an exceptionally large or small feature value in comparison with other training objects. In general, trained classifiers or regressors only provide reliable estimates for input data close to the training set. Extrapolations to unknown and remote regions in feature space are very uncertain (Roberts & Penny, 1996). Neural networks, for instance, can be trained to estimate posterior probabilities (Richard & Lippmann, 1991; Bishop, 1995; Ripley, 1996) and tend to give high confidence outputs for objects which are remote from the training set. In these cases outlier detection should be used first to detect and reject outliers to avoid unfounded confident classifications.

Secondly, data description can be used for a classification problem where one of the classes is sampled very well, while the other class is severely undersampled. An example is a machine monitoring system in which the current condition of a machine is examined. An alarm is raised when the machine shows a problem. Measurements on the normal working conditions of a machine are very cheap and easy to obtain. Measurements of outliers, on

the other hand, would require the destruction of the machine in all possible ways. It is very expensive, if not impossible, to generate all faulty situations (Japkowicz, Myers, & Gluck, 1995). Only a method which uses mainly the target data, and does not require representative outlier data, can solve this monitoring problem.

The last possible use of outlier detection is the comparison of two data sets. Assume a classifier has been trained (by a long and difficult optimization) on some (possibly expensive) data. When a similar problem has to be solved, the new data set can be compared with the old training set. In case of incomparable data, the training of a new classifier will be needed.

In many one-class classification problems an extra complication occurs, namely that it is beforehand not clear what the specific distribution of the data will be in practice. The operator of the machine can easily run the machine in different, but legal modes, covering the complete operational area of the machine. Although it defines the normal working area, the distribution over this area is not to be trusted. In practice the machine may stay much longer in one mode than in another, and this mode might have been sampled sparsely during the training phase. A data description for this type of data should therefore model the boundary of the normal class, instead of modeling the complete density distribution.

Several solutions for solving the data description problem have been proposed. Most often the methods focus on outlier detection. Conceptually the simplest solution for outlier detection is to generate outlier data around the target set. An ordinary classifier is then trained to distinguish between the target data and outliers (Roberts et al., 1994). Koch et al. (1995) used ART-2A and a Multi-Layered Perceptron for the detection of (partially obscured) objects in an automatic target recognition system. Unfortunately this method requires the availability of a set of near-target objects (possibly artificial) belonging to the outlier class. The methods scale very poorly in high dimensional problems, especially when the near-target data has to be created and is not readily available.

In classification or regression problems a more advanced Bayesian approach can be used for detecting outliers (Bishop, 1995; MacKay, 1992; Roberts & Penny, 1996). Instead of using the most probable weight configuration of a classifier (or regressor) to compute the output, the output is weighted by the probability that the weight configuration is correct given the data. This method can then provide an estimate of the probability for a certain object given the model family. Low probabilities will then indicate a possible outlier. The classifier outputs are moderated automatically for objects remote from the training domain. These methods are not optimized for outlier detection though; they require a classification (or regression) task to be solved and can be computationally expensive.

Most often the task of data description is solved by estimating a probability density of the target data (Barnett & Lewis, 1978). For instance in Bishop (1994) and Tarassenko, Hayton, and Brady (1995) the density is estimated by a Parzen density estimator, whereas in Parra, Deco, and Miesbach (1996) one Gaussian distribution is used. In (Ritter & Gallegos, 1997) not only the target density is estimated, but also the outlier density. The first drawback is that in general (in higher dimensional feature spaces) a large number of samples is required. The second drawback is that they will not be very resistant to training data which only defines the area of the target data, and which does not represent the complete density distribution. It will mainly focus on modeling the high density areas, and reject low density areas, although they may define legal target data.

Vapnik argued that in order to solve a problem, one should not try to solve a more general problem as an intermediate step (Vapnik, 1998). The estimation of the complete density instead of computing the boundary around a data set might require too much data and could result in bad descriptions. An attempt to train just the boundaries of a data set is made in Moya and Hush (1996) and Moya, Koch, and Hostetler (1993). Here neural networks are trained with extra constraints to give closed boundaries. Unfortunately this method inherits the weak points in neural network training, i.e. the choice of the size of the network, weight initialization, the stopping criterion, etc. Rosen (1965) made a data description as a by-product of a classification problem. He shows how the classification problem can be formulated as a convex programming problem. When the classification problem is not linearly separable, an ellipsoidal separation can be applied, where one of the classes is enclosed by an ellipsoid of minimum size. We will use a similar method for the one-class classification problem.

Finally Schölkopf et al. used an hyperplane (Schölkopf et al., 1999b) to separate the target objects from the origin with maximal margin. This formulation is comparable with the Support Vector Classifier by Vapnik (1998) and it is possible to define implicit mappings to obtain more flexible descriptions.

In this paper we discuss a method which obtains a spherically shaped boundary around the complete target set with the same flexibility. To minimize the chance of accepting outliers, the volume of this description is minimized. We will show how the outlier sensitivity can be controlled by changing the ball-shaped boundary into a more flexible boundary and how example outliers can be included into the training procedure (when they are available) to find a more efficient description.

In Section 2 we present the basic theory, which is presented in part in Tax and Duin (1999). The normal data description, and also the description using negative examples will be explained and we will compare the method with the hyperplane approach in Schölkopf et al. (1999b). In Section 3 some basic characteristics of this data description will be shown, and in Section 4 the method will be applied to a real life problem and compared with some other density estimation methods. Section 5 contains some conclusions.

2. Theory

To begin, we fix some notation. We assume vectors $\mathbf{x}$ are column vectors and $\mathbf{x}^2 = \mathbf{x} \cdot \mathbf{x}$. We have a training set $\{\mathbf{x}_i\}, i = 1, \ldots, N$ for which we want to obtain a description. We further assume that the data shows variances in all feature directions.

2.1. Normal data description

To start with the normal data description, we define a model which gives a closed boundary around the data: an hypersphere. The sphere is characterized by center $\mathbf{a}$ and radius $R > 0$. We minimize the volume of the sphere by minimizing R^2, and demand that the sphere contains all training objects $\mathbf{x}_i$. This is identical to the approach which is used in Schölkopf, Burges, and Vapnik (1995) to estimate the VC-dimension of a classifier (which is bounded by the diameter of the smallest sphere enclosing the data). At the end of this section we will

show that this approach gives solutions similar to the hyperplane approach of Schölkopf et al. (1999b).

Analogous to the Support Vector Classifier (Vapnik, 1998) we define the error function to minimize:

$$F(R, \mathbf{a}) = R^2 \tag{1}$$

with the constraints:

$$\|\mathbf{x}_i - \mathbf{a}\|^2 \leq R^2, \quad \forall i \tag{2}$$

To allow the possibility of outliers in the training set, the distance from $\mathbf{x}_i$ to the center $\mathbf{a}$ should not be strictly smaller than R^2, but larger distances should be penalized. Therefore we introduce slack variables $\xi_i \geq 0$ and the minimization problem changes into:

$$F(R, \mathbf{a}) = R^2 + C \sum_i \xi_i \tag{3}$$

with constraints that almost all objects are within the sphere:

$$\|\mathbf{x}_i - \mathbf{a}\|^2 \leq R^2 + \xi_i, \quad \xi_i \geq 0 \quad \forall i \tag{4}$$

The parameter C controls the trade-off between the volume and the errors.

Constraints (4) can be incorporated into Eq. (3) by using Lagrange multipliers:

$$\begin{aligned} L(R, \mathbf{a}, \alpha_i, \gamma_i, \xi_i) = R^2 + C \sum_i \xi_i \\ - \sum_i \alpha_i \{R^2 + \xi_i - (\|\mathbf{x}_i\|^2 - 2\mathbf{a} \cdot \mathbf{x}_i + \|\mathbf{a}\|^2)\} - \sum_i \gamma_i \xi_i \end{aligned} \tag{5}$$

with the Lagrange multipliers $\alpha_i \geq 0$ and $\gamma_i \geq 0$. L should be minimized with respect to $R, \mathbf{a}, \xi_i$ and maximized with respect to α_i and γ_i.

Setting partial derivatives to zero gives the constraints:

$$\frac{\partial L}{\partial R} = 0 : \quad \sum_i \alpha_i = 1 \tag{6}$$

$$\frac{\partial L}{\partial \mathbf{a}} = 0 : \quad \mathbf{a} = \frac{\sum_i \alpha_i \mathbf{x}_i}{\sum_i \alpha_i} = \sum_i \alpha_i \mathbf{x}_i \tag{7}$$

$$\frac{\partial L}{\partial \xi_i} = 0 : \quad C - \alpha_i - \gamma_i = 0 \tag{8}$$

From the last equation $\alpha_i = C - \gamma_i$ and because $\alpha_i \geq 0$, $\gamma_i \geq 0$, Lagrange multipliers γ_i can be removed when we demand that

$$0 \leq \alpha_i \leq C \tag{9}$$

Resubstituting (6)–(8) into (5) results in:

$$L = \sum_i \alpha_i (\mathbf{x}_i \cdot \mathbf{x}_i) - \sum_{i,j} \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j) \tag{10}$$

subject to constraints (9). Maximizing (10) gives a set α_i. When an object $\mathbf{x}_i$ satisfies the inequality $\|\mathbf{x}_i - \mathbf{a}\|^2 < R^2 + \xi_i$, the constraint is satisfied and the corresponding Lagrange multiplier will be zero ($\alpha_i = 0$). For objects satisfying the equality $\|\mathbf{x}_i - \mathbf{a}\|^2 = R^2 + \xi_i$ the constraint has to be enforced and the Lagrange multiplier will become unequal zero ($\alpha_i > 0$).

$$\|\mathbf{x}_i - \mathbf{a}\|^2 < R^2 \rightarrow \alpha_i = 0, \gamma_i = 0 \tag{11}$$

$$\|\mathbf{x}_i - \mathbf{a}\|^2 = R^2 \rightarrow 0 < \alpha_i < C, \gamma_i = 0 \tag{12}$$

$$\|\mathbf{x}_i - \mathbf{a}\|^2 > R^2 \rightarrow \alpha_i = C, \gamma_i > 0 \tag{13}$$

Equation (7) shows that the center of the sphere is a linear combination of the objects. Only objects $\mathbf{x}_i$ with $\alpha_i > 0$ are needed in the description and these objects will therefore be called the *support vectors* of the description (SV's).

To test an object $\mathbf{z}$, the distance to the center of the sphere has to be calculated. A test object $\mathbf{z}$ is accepted when this distance is smaller or equal than the radius:

$$\|\mathbf{z} - \mathbf{a}\|^2 = (\mathbf{z} \cdot \mathbf{z}) - 2 \sum_i \alpha_i (\mathbf{z} \cdot \mathbf{x}_i) + \sum_{i,j} \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j) \leq R^2 \tag{14}$$

By definition, R^2 is the distance from the center of the sphere $\mathbf{a}$ to (any of the support vectors on) the boundary. Support vectors which fall outside the description ($\alpha_i = C$) are excluded. Therefore:

$$R^2 = (\mathbf{x}_k \cdot \mathbf{x}_k) - 2 \sum_i \alpha_i (\mathbf{x}_i \cdot \mathbf{x}_k) + \sum_{i,j} \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j) \tag{15}$$

for any $\mathbf{x}_k \in SV_{<C}$, the set of support vectors which have $\alpha_k < C$.

Note that in all formulae (Eqs. (10), (14) and (15)) objects $\mathbf{x}_i$ only appear in the form of inner products with other objects $(\mathbf{x}_i \cdot \mathbf{x}_j)$. Analogous to Vapnik (1998) the inner products can be replaced by a kernel function to obtain more flexible methods. We will use this in Section 2.3.

The left plot in figure 1 shows an example description for a small 2 dimensional banana-shaped data set. For this description $C = 1$ is taken, indicating the hard-margin solution (by constraint (6) the upper bound constraint on α_i in (9) will always be satisfied for $C \geq 1$). The solid circles indicate the support vectors, the dashed line is the boundary of the data description. The gray value indicates the distance to the center of the sphere; dark is close, light is remote. Only three objects are required to describe the complete data set.

Schölkopf gives an alternative approach for solving the data description problem (Schölkopf et al., 1999b). A hyperplane is placed such that it separates the dataset from

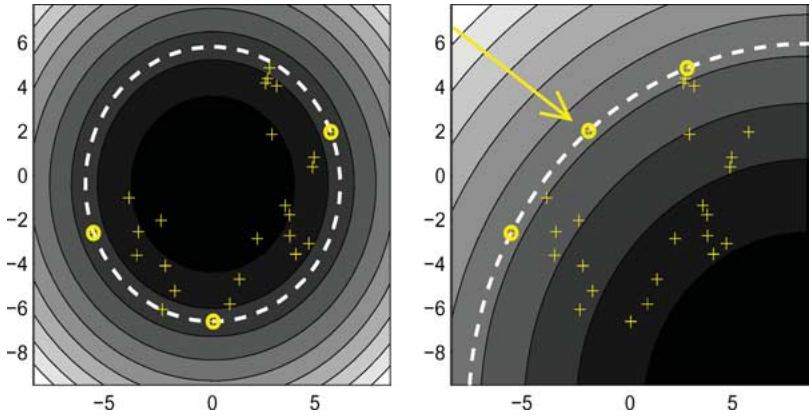

Figure 1. Example of a data description without outliers (left) and with one outlier (right).

the origin with maximal margin. Although this is not a closed boundary around the data, it gives comparable solutions when the data is preprocessed to have unit norm (see also Schölkopf et al., 1999a).

For hyperplane $\mathbf{w}$ which separates the data $\mathbf{x}_i$ from the origin with margin ρ, the following holds:

$$\mathbf{w} \cdot \mathbf{x}_i \geq \rho - \xi_i \quad \forall i \quad \xi_i \geq 0 \tag{16}$$

where ξ_i accounts for possible errors. Schölkopf minimizes the structural error of the hyperplane, measured by $\|\mathbf{w}\|$. This results in the following minimization problem:

$$\min_{\mathbf{w},\rho,\xi} \frac{1}{2}\|\mathbf{w}\|^2 - \rho + \frac{1}{\nu N}\sum_i \xi_i \tag{17}$$

with constraints (16). The regularization parameter $\nu \in (0, 1)$ is a user defined parameter indicating the fraction of the data that should be separated and can be compared with the parameter C in the SVDD. Here we will call this method the ν-SVC.

An equivalent formulation of (16) and (17) is

$$\max_{\mathbf{w},\rho,\xi} \rho - \frac{1}{\nu N}\sum_i \xi_i, \quad \text{with} \quad \mathbf{w} \cdot \mathbf{x}_i \geq \rho - \xi_i, \quad \xi_i \geq 0, \quad \|\mathbf{w}\| = 1 \tag{18}$$

where a constraint on $\|\mathbf{w}\|$ is introduced. When all the data in the original SVDD formulation is transformed to unit norm (see also (3) and (4)), we obtain the following optimization problem:

$$\min\ R'^2 + C' \sum_i \xi'_i \quad \text{with} \quad \|\mathbf{x}'_i - \mathbf{a}'\|^2 \leq R'^2 + \xi'_i \quad \forall i \tag{19}$$

where $\mathbf{x}'$ and $\mathbf{a}'$ are normalized vectors. Rewriting gives:

$$\max -R'^2 - C' \sum_i \xi_i' \quad \text{with} \quad 2(\mathbf{a}' \cdot \mathbf{x}_i') \geq 2 - R'^2 - \xi_i' \tag{20}$$

We define $\mathbf{w} = 2\mathbf{a}'$, $\rho = 2 - R'^2$, $\frac{1}{\nu N} = C'$, $\xi_i = \xi_i'$ and the following optimization problem is obtained:

$$\max \ -2 + \rho - \frac{1}{\nu N} \sum_i \xi_i \quad \text{with} \quad \mathbf{w} \cdot \vec{\mathbf{x}}_i \geq \rho - \xi_i, \quad \xi_i \geq 0, \quad \|\mathbf{w}\| = 2 \tag{21}$$

which is a comparable solution to Eq. (18). It differs in the constraint on the norm of $\mathbf{w}$ and in an offset of 2 in the error function. For non-normalized data the solutions become incomparable due to the difference in model of the description (hyperplane or hypersphere). We will come back to the ν-SVC in Section 2.3 in the discussion on kernels.

2.2. *SVDD with negative examples*

When negative examples (objects which should be rejected) are available, they can be incorporated in the training to improve the description. In contrast with the training (target) examples which should be within the sphere, the negative examples should be *outside* it. This data description now differs from the normal Support Vector Classifier in the fact that the SVDD always obtains a closed boundary around one of the classes (the target class). The support vector classifier just distinguishes between two (or more) classes and cannot detect outliers which do not belong to any of the classes.

In the following the target objects are enumerated by indices i, j and the negative examples by l, m. For further convenience assume that target objects are labeled $y_i = 1$ and outlier objects are labeled $y_l = -1$. Again we allow for errors in both the target and the outlier set and introduce slack variables ξ_i and ξ_l:

$$F(R, \mathbf{a}, \xi_i, \xi_l) = R^2 + C_1 \sum_i \xi_i + C_2 \sum_l \xi_l \tag{22}$$

and the constraints

$$\|\mathbf{x}_i - \mathbf{a}\|^2 \leq R^2 + \xi_i, \quad \|\mathbf{x}_l - \mathbf{a}\|^2 \geq R^2 - \xi_l, \quad \xi_i \geq 0, \ \xi_l \geq 0 \quad \forall i, l \tag{23}$$

These constraints are again incorporated in Eq. (22) and the Lagrange multipliers α_i, α_l, γ_i, γ_l are introduced:

$$\begin{aligned} L(R, \mathbf{a}, \xi_i, \xi_l, \alpha_i, \alpha_l, \gamma_i, \gamma_l) &= R^2 + C_1 \sum_i \xi_i + C_2 \sum_l \xi_l - \sum_i \gamma_i \xi_i - \sum_l \gamma_l \xi_l \\ &- \sum_i \alpha_i [R^2 + \xi_i - (\mathbf{x}_i - \mathbf{a})^2] - \sum_l \alpha_l [(\mathbf{x}_l - \mathbf{a})^2 - R^2 + \xi_l] \end{aligned} \tag{24}$$

with $\alpha_i \geq 0, \alpha_l \geq 0, \gamma_i \geq 0, \gamma_l \geq 0$.

Setting the partial derivatives of L with respect to R, $\mathbf{a}$, ξ_i and ξ_l to zero gives the constraints:

$$\sum_i \alpha_i - \sum_l \alpha_l = 1 \tag{25}$$

$$\mathbf{a} = \sum_i \alpha_i \mathbf{x}_i - \sum_l \alpha_l \mathbf{x}_l \tag{26}$$

$$0 \leq \alpha_i \leq C_1, \quad 0 \leq \alpha_l \leq C_2 \quad \forall i, l \tag{27}$$

When Eqs. (25)–(27) are resubstituted into Eq. (24) we obtain

$$L = \sum \alpha_i (\mathbf{x}_i \cdot \mathbf{x}_i) - \sum_l \alpha_l (\mathbf{x}_l \cdot \mathbf{x}_l) - \sum_{i,j} \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j) \tag{28}$$

$$+ 2 \sum_{l,j} \alpha_l \alpha_j (\mathbf{x}_l \cdot \mathbf{x}_j) - \sum_{l,m} \alpha_l \alpha_m (\mathbf{x}_l \cdot \mathbf{x}_m) \tag{29}$$

When we finally define new variables $\alpha'_i = y_i \alpha_i$ (index i now enumerates both target and outlier objects), the SVDD with negative examples is identical to the normal SVDD. The constraints given in Eqs. (25) and (26) change into $\sum_i \alpha'_i = 1$ and $\mathbf{a} = \sum_i \alpha'_i \mathbf{x}_i$ and again the testing function Eq. (14) can be used. Therefore, when outlier examples are available, we will replace Eq. (10) by (29) and we will use α'_i instead of α_i.

In the right plot in figure 1 the same data set is shown as in the left plot, extended with one outlier object (indicated by the arrow). The outlier lies within the original description on the left. A new description has to be computed to reject this outlier. With a minimal adjustment to the old description, the outlier is placed on the boundary of the description. It becomes a support vector for the outlier class and cannot be distinguished from the support vectors from the target class on the basis of Eq. (14). Although the description is adjusted to reject the outlier object, it does not fit tightly around the rest of the target set anymore. A more flexible description is required.

2.3. Flexible descriptions

When instead of the rigid hypersphere a more flexible data description is required, another choice for the inner product $(\mathbf{x}_i \cdot \mathbf{x}_j)$ can be considered. Replacing the new inner product by a kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j))$ an implicit mapping Φ of the data into another (possibly high dimensional) feature space is defined. An ideal kernel function would map the target data onto a bounded, spherically shaped area in the feature space and outlier objects outside this area. Then the hypersphere model would fit the data again (this is comparable with replacing the inner product in the Support Vector classifier when the classes are not linearly separable). Several kernel functions have been proposed for the Support Vector Classifier (Vapnik, 1998; Smola, Schölkopf, & Müller, 1998). It appears that not all these kernel functions map the target set in a bounded region in feature space. To show this, we first investigate the polynomial kernel.

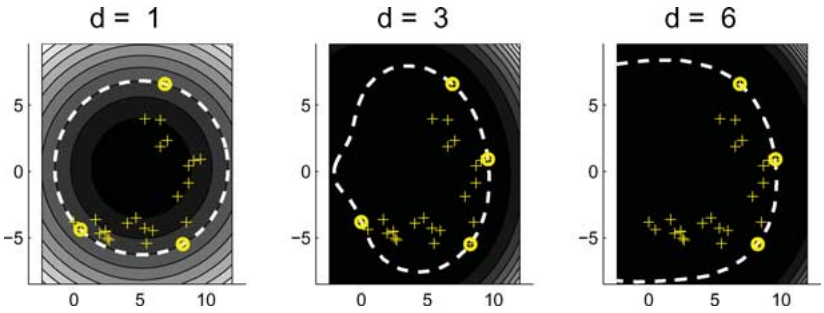

Figure 2. Data description trained on a banana shaped data set. Kernel is a polynomial kernel with varying degrees. Support vectors are indicated by the solid circles, the dashed line is the description boundary.

The polynomial kernel is given by

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)^d \tag{30}$$

where the free parameter $d \in \mathbb{N}^+$ gives the degree of the polynomial kernel. The testing function of the SVDD (Eq. (14)) shows that only the second term accounts for the interaction between the test object $\mathbf{z}$ and the support objects $\mathbf{x}_i$. Recall that $\mathbf{x}_i \cdot \mathbf{x}_j = \cos(\theta_{ij})\|\mathbf{x}_i\| \cdot \|\mathbf{x}_j\|$ where θ_{ij} is the angle between object vector $\mathbf{x}_i$ and $\mathbf{x}_j$. When data is not centered around the origin, object vectors can become large. And when the θ_{ij} is small, $\cos(\theta_{ij}) \sim 1$ will stay almost constant. Then for larger degrees d, the polynomial kernel is approximated by:

$$(\mathbf{x}_i \cdot \mathbf{x}_j)^d = \cos^d(\theta_{ij})\|\mathbf{x}_i\|^d \cdot \|\mathbf{x}_j\|^d \simeq \|\mathbf{x}_i\|^d \cdot \|\mathbf{x}_j\|^d \tag{31}$$

Equation (31) looses the sensitivity to θ_{ij} in the neighborhood of the training data (where θ becomes small). The objects with the largest norm in the training set will overwhelm all other terms in the polynomial kernel. This effect can be suppressed by centering the data around the origin and rescaling the data to unit standard variance. Unfortunately rescaling to unit variance might only magnify the noise in directions with small variance, and the influence of the norms is not avoided. Finally, centering the data in the feature space by subtracting the averaged $\mathbf{x}$ (as explained in Schölkopf, 1997) does not resolve the problem of the large differences in vector norms. It can be shown that the centered SVDD is equivalent to the original SVDD.

The influence of large norm objects is visible in figure 2. For a simple 2 dimensional data set, descriptions are obtained using a polynomial kernel with different degrees, ranging from $d = 1.0$ (left) to $d = 6.0$ (right). Again the solid circles indicate the support vectors, the dashed line is the description boundary mapped in the input space. The rigid spherical description is obtained for $d = 1.0$. For degree $d = 6.0$ the description is a sixth order polynomial. Here the training objects most remote from the origin (the objects on the right) become support objects and the data description only distinguishes on the basis of the norm of the vectors. Large regions in the input space without target objects will be accepted by the description.

Next we investigate the Gaussian kernel with

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/s^2) \tag{32}$$

This kernel is independent of the position of the data set with respect to the origin, it only utilizes the distances between objects. In the testing function (Eq. (14)) the first term equals 1.0 and the testing function boils down to a weighted sum of Gaussians. Test object $\mathbf{z}$ is accepted when:

$$\sum_i \alpha_i \exp\left(\frac{-\|\mathbf{z} - \mathbf{x}_i\|^2}{s^2}\right) \geq -R^2/2 + C_R \tag{33}$$

where C_R depends only on the Support Vectors $\mathbf{x}_i$ and not on $\mathbf{z}$. Using the Gaussian kernel the influence of the norms of the objects is avoided. The objects are mapped to unit norm vectors (the norm of the mapped objects $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_i) = 1$) and only the angles between object vectors count. When Gaussian kernels are used, both the SVDD and the ν-SVC give comparable solutions.

For small values of s, $\exp(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{s^2}) \simeq 0, \forall i \neq j$ and Eq. (10) is optimized when all objects become support objects with equal $\alpha_i = \frac{1}{N}$. This is identical to the Parzen density estimation with a small kernel width. For very large s the solution approximates the original spherically shaped solution. This can be shown by a Taylor expansion of the Gaussian kernel:

$$K(\mathbf{x}_i, \mathbf{x}_j) = 1 - \|\mathbf{x}_i\|^2/s^2 - \|\mathbf{x}_j\|^2/s^2 + 2(\mathbf{x}_i \cdot \mathbf{x}_j)/s^2 + \cdots \tag{34}$$

Substituting Eq. (34) into Eq. (10) we obtain:

$$\begin{aligned} L &= \sum_i \alpha_i (1 - \|\mathbf{x}_i\|^2/s^2 - \|\mathbf{x}_i\|^2/s^2 + 2(\mathbf{x}_i \cdot \mathbf{x}_i)/s^2 + \cdots) \\ &\quad - \sum_{i,j} \alpha_i \alpha_j (1 - \|\mathbf{x}_i\|^2/s^2 - \|\mathbf{x}_j\|^2/s^2 + 2(\mathbf{x}_i \cdot \mathbf{x}_j)/s^2 + \cdots) \\ &= 1 - 1 + 2\sum \alpha_i \|\mathbf{x}_i\|^2/s^2 - 2\sum_{i,j} \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j)/s^2 + \cdots \end{aligned} \tag{35}$$

Ignoring higher orders this is equivalent to Eq. (10) (with an extra scaling factor $2/s^2$). When a new regularization parameter $C' = (2C)/s^2$ is used, an identical solution is found.

For intermediate values of s a weighted Parzen density estimation is obtained. Both the weights of the kernels and the choice of which training objects become support vectors are obtained automatically by the optimization procedure.

These situations, ranging from a Parzen density estimation to the rigid hyper sphere, can be observed in figure 3, in combination with varying values of C. In the left column of pictures a small $s = 1.0$ width parameter is used, and in the right column a large $s = 15$ is used. In all cases, except for the limiting case where s becomes huge, the description is tighter than the normal spherically shaped description or the description with the polynomial kernel. Note that with increasing s the number of support vectors decreases. Secondly,

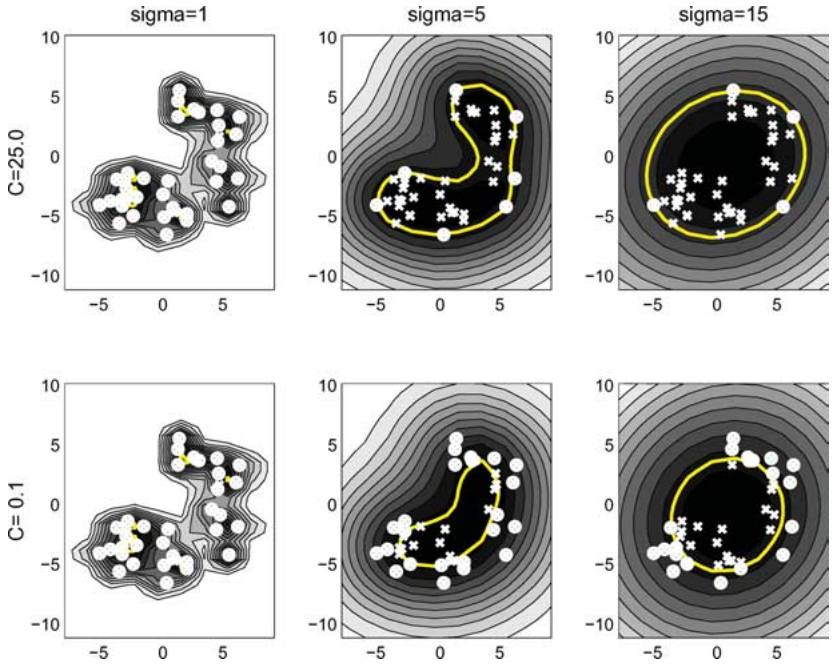

Figure 3. Data description trained on a banana shaped data set. A Gaussian kernel with different widths ($s = 1, 5, 15$) and different values for C ($C = 40$, almost the hard margin case, and $C = 0.1$) are used. Support vectors are indicated by the solid circles, the solid white line is the description boundary.

decreasing the parameter C constraints the values for α_i more, and more objects become support vector. The error on the target class increases, but the covered volume of the data description decreases with decreasing C.

2.4. Target error estimate

When an object drawn from the target distribution is rejected by the description, it is called an error. By applying Leave-One-Out estimation (Vapnik, 1998; Bishop, 1995), it can be shown that the number of support vectors is an indication of the expected error made on the target set. For that the notion of *essential support vectors* has to be introduced (Vapnik, 1998). The expansion of the center of the description $\mathbf{a} = \sum_i \alpha_i \mathbf{x}_i$ is not unique. It might be that more objects are on the boundary of the sphere than is necessary for the description (for instance when four objects are on a circle in a 2 dimensional feature space where 3 are sufficient[1]). The essential support vectors are these objects which appear in all possible expansions.

When one of the internal points ($\alpha_i = 0$) is left out of the training and the data description is computed, the same solution is obtained as with the training set including this training object. During testing this object will therefore be accepted by the description. When a non-essential support vector is left out during training, the solution including this object is still

obtained, using the remaining support vectors in the expansion. When an essential support object on the boundary (a support vector with $\alpha_i < C$) is left out, a smaller description is found. This support point will be rejected by the new solution. The errors, the support objects with $\alpha_i = C$ are already outside the sphere. When they are left out of the training set, they will again be rejected. The fraction of objects which become (essential) support objects and outlier, is thus an leave-one-out error estimate on the target set:

$$\tilde{E}_{\text{LOO}} \leq \frac{\#\, SVs + \#\,\text{errors}}{N} \tag{36}$$

(where the inequality holds when not all support vectors are essential support vectors).

When the Support Vector Data Description is trained on a set with both target and outlier objects present, only the fraction of support vectors on the target set should be considered for the estimation of the leave-one out error. The error of the outlier class can also be estimated, but only when a representative set of outliers is drawn from the true outlier distribution. Because usually in the one-class classification problem the outlier class is sampled very sparsely, a good representative data set is often not available and a reliable estimate of the error is not possible.

This target error estimate opens the possibility to optimize the regularization parameter C and the width s of the Gaussian kernel. The value of the regularization parameter C can be chosen by using the fact that for errors $\alpha_i = C$. Recall the constraint that $\sum_i \alpha_i = 1$ (or $\sum_i \alpha_i'$ when negative examples are available) and therefore $\frac{\#\,\text{errors}}{N} \leq \frac{1}{NC}$. When in a training set no errors are expected C can be set to 1.0, indicating that all target data should be accepted and all negative examples should be rejected. When no negative examples are available and some outliers in the training set are expected, set $C \leq \frac{1}{N \cdot (\text{fraction outlier})}$ beforehand. In Schölkopf et al. (1999a) the parameter (fraction outlier) is called ν and the fact that this ν sets the error (here on the target class) is called the ν-property.

Secondly the kernel width s can be optimized on the basis of the required target acceptance rate. Comparing two width values $s_1 > s_2$, we get $\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/s_1^2) > \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/s_2^2)$. For large enough s_1 Eq. (10) is maximized when the α_i, α_j corresponding to the larger terms $K(\mathbf{x}_i, \mathbf{x}_j)$ become zero. This means that the most separate objects become support vector and that the number of support vectors tends to decrease with increasing s. Using Eq. (36) this also means a decrease of the expected error on the target set (this should be no surprise, because the covered area in feature space tends to increase when the data model changes from a Parzen density estimate with small σ to the rigid hypersphere, see again figure 3). Unfortunately the minimization of Eq. (10) with Gaussian kernel requires s beforehand, so the optimization requires an iterative scheme to minimize the difference between the fraction SV's and the user defined error fraction.

When a small fraction of the target data is to be rejected, the boundary is located in the tails of the target distribution. In order to have reliable estimates in these boundary in the tails, the required sample sizes become large. Consider a dataset with 10 objects in a 2 dimensional data set. In a typical training of an SVDD, it appears that the minimum number of support vectors is 2 to 3. To obtain a target error lower than 20% on this data, more training data is required. The accuracy of the target error estimate, the required number of training objects and some other characteristics will be investigated in the next section.

3. SVDD characteristics

In this section several characteristics of the SVDD will be investigated. In all experiments the Gaussian kernel is used and an upper bound on the error on the target set is set a priori and the width parameter s is optimized.

3.1. Number of support vectors

The question of how much data is required for finding a sufficiently accurate description (in terms of classification error) of a target class in some feature space cannot be answered a priori. Not only does it depend on the shape of the target data set, it also depends on the distribution of the (unknown) outlier set. However, a lower limit can be obtained, when a rigid spherically shaped data description is considered. Here just the coordinates of the center of the sphere and the radius of the sphere are required. Therefore in theory only two objects are sufficient to determine the sphere (independent of dimensionality). The weights α_i are positive and constrained to sum up to 1 (Eq. (6)), and therefore the sphere center has to be within the convex hull of the support vectors. The center can therefore only be described by two objects when it is located on the line connecting these objects. For d-dimensional data which has variation in all directions, the required number of objects can increase up to $d + 1$. For data sets in a subspace, the number becomes less (and down to two if the data is on a one dimensional line).

In figure 4 the fraction of outliers which is accepted, the fraction of training objects which become support vectors and the error on the target set is shown for data with different dimensionalities, 2D and 5D. The data is drawn from the (artificially generated) banana shaped distribution. For varying s an SVDD is trained. The error on the target set is then estimated by drawing a new independent test set containing 200 objects. 1000 outliers are drawn from a square block around the data.

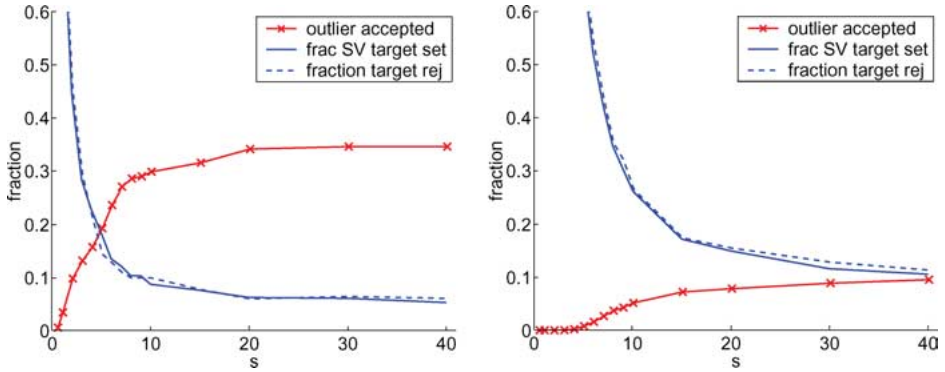

Figure 4. The fraction of (independent test) outliers which is accepted, the fraction of the (training) target data which becomes support vector and the fraction of (independent test) target data rejected vs. the width parameter s for (left) 2 dimensional and (right) 5 dimensional data. The training set size is 50.

By increasing s the number of support vectors decrease, closely followed by the error on the target set. For large s the error is bounded by the minimum number of support vectors which is needed for the description, in this case about 3 support vectors for 2D data and about 5 for 5D data. For a training set size of 50 this results in a minimum error of about 6% or 10%. The maximal fraction of outliers accepted is the quotient between the volume of the hypersphere and the volume of the outlier data block. For $s > 20$ in the 2D data and $s > 40$ in the 6D data both fractions stay constant, indicating that the maximum scale in the data is reached.

Note that for increasing dimensionality of the feature space the volume of the outlier block tends to grow faster than the volume of the target class. This means that the overlap between the target and outlier data decreases and the classification problem becomes easier. In practice, this phenomenon is less apparent, although still present, because outliers are often not completely randomly scattered in all feature directions. Increasing the dimensionality to improve classification performance does have it's limits, because with increasing the dimensionality also the target class boundary increases, such that for reliable estimation of the boundary, more data is required.

In Table 1 the minimum number of support vectors is shown for different datasets and varying dimensionalities (from 2D to the full dimensionality, using all available features). The first three datasets, gauss, banana and ellipse, are artificially generated. The Gauss and

Table 1. Minimum number of support vectors, for different datasets (with dimensionalities from 2D to the full dimensionality) and different target classes.

Dataset	Full	2D	5D	10D	25D
Gauss		2.94 (0.64)	5.63 (0.97)	7.81 (1.69)	11.42 (2.06)
Banana		3.23 (0.71)	5.50 (1.48)	7.76 (2.61)	8.71 (5.36)
Ellipse		2.08 (0.04)	2.16 (0.08)	2.17 (0.07)	2.34 (0.19)
Iris	2.00 (0.00)	2.00 (0.00)			
(4D)	2.20 (0.63)	2.10 (0.32)			
	2.30 (0.67)	2.10 (0.32)			
Sonar	6.20 (1.23)	2.40 (0.97)	2.70 (1.34)	5.60 (0.97)	6.00 (1.33)
(60D)	7.30 (1.70)	3.10 (0.32)	4.40 (0.97)	4.90 (1.37)	7.20 (1.40)
Dataset	Full	2D	5D	10D	25D
Glass	3.90 (0.32)	3.00 (0.67)	3.90 (0.32)		
(9D)	3.00 (0.47)	2.10 (0.32)	2.30 (0.67)		
	2.30 (0.48)	2.20 (0.42)	2.20 (0.42)		
	4.20 (0.42)	3.70 (0.48)	4.30 (0.48)		
Imox	4.00 (0.47)	3.80 (0.42)	4.00 (0.47)		
(8D)	2.60 (1.07)	2.30 (0.67)	2.60 (1.07)		
	3.70 (0.67)	2.20 (0.42)	3.10 (0.57)		
	5.00 (0.82)	3.30 (0.67)	5.00 (0.82)		

Values are averaged of 10 runs, the standard deviation is given between brackets.

banana set have variance in all feature directions, the ellipse mainly in the first direction. The other datasets are normal classification problems, taken from the UCI repository (Blake, Keogh, & Merz, 1998). A data description is trained on each of the classes separately (thus the three class Iris dataset shows three lines of results). In the first column the number of SV's in the original feature space is shown (except for the artificial datasets). In the next columns it is shown for datasets which are reduced to the first few principal components.

For the Gauss and banana datasets the number of SV's increases with increasing dimensionality, but stays in the order of $d/2$. In the ellipse dataset the single large variance direction causes this number to stay very low (two support vectors on the extremities of the dataset often support the whole data description). For the other data sets the number of support vectors is mainly determined by the dimensionality of the subspace in which the data is distributed. For the sonar database the number of SV's hardly changes after 25D, for the other datasets this already happens with 5D data. Finally note that the number of SV's for different classes can differ, e.g. in imox this number ranges from 2 to almost 5. When this limited amount of data is available, this minimum number of support vectors immediately gives an indication of the target error which can minimally be achieved (by Eq. (36)). Only using more training data can reduce the error on the target set further.

3.2. Training with outliers

To compare data descriptions trained with outlier data with standard two-class classifiers, results on the classification problems mentioned in the previous section are shown in Table 2. One class is the target class, and all other data is outlier data. Tenfold cross validation is used to find the classification error. The reported results are the classification errors on an independent (labeled) test set. The classifiers are Gaussian-density based linear classifier (called Bayes), Parzen classifier and the Support Vector Classifier with polynomial kernel, degree 3. These are compared with the data descriptions, without and including example outliers (SVDD and SVDD-neg respectively). In both these cases the Gaussian kernel was used. The parameter σ was optimized such that approximately 10% of the data becomes support vector. When example outliers are used, the parameters C_1 and C_2 are set such, that the fraction of rejected objects on the target set should be less then 1%, while the fraction of accepted outlier objects can be up to 50%.

To investigate the influence of the norms of the vectors in real world applications, not only the Gaussian kernel but also the polynomial kernel (with degree 3, where the data is rescaled to unit variance) is used. These results are given in the columns SVDD,p3 and SVDD-neg, p3.

In most cases the classifiers, especially the Parzen classifier and the SVC with polynomial degree 3, outperform the data descriptions. This is not surprising, because they are constructed to give the best separation between the two classes, with equal focus on each of the classes. They are not limited by the extra constraint to have a closed boundary around on the target class.

The performance of the one-class classifiers which only use information from the target set, perform worse, but in some cases still comparable to the classifiers which use information of both the target and outlier data. The data descriptions using outliers perform

Table 2. Cross-validation error (in %) of different classifiers for distinguishing between one of the classes and the rest.

cl nr	Bayes	Parzen	SVC p3	SVDD	SVDD p3	SVDD neg	SVDD neg, p3
				Iris			
1	**0.0 (0.0)**	**0.0 (0.0)**	**0.0 (0.0)**	**4.7 (6.3)**	33.3 (0.0)	8.0 (6.1)	**4.7 (4.5)**
2	26.7 (9.4)	**3.3 (3.5)**	8.0 (6.1)	**8.0 (8.8)**	23.3 (8.5)	**9.3 (4.7)**	10.0 (3.5)
3	7.3 (4.9)	**3.3 (3.5)**	6.0 (3.8)	**7.3 (3.8)**	38.7 (8.2)	**9.3 (5.6)**	20.0 (4.4)
				Glass			
1	21.9 (8.6)	**19.2 (7.8)**	22.4 (8.7)	**33.6 (7.8)**	36.5 (9.3)	**29.0 (5.7)**	32.7 (8.6)
2	32.6 (10.2)	**19.2 (6.9)**	20.5 (11.2)	51.5 (5.1)	**50.6 (6.1)**	**30.4 (7.9)**	51.9 (7.0)
3	**8.4 (3.0)**	**8.4 (3.7)**	10.3 (4.4)	**30.0 (10.9)**	54.7 (6.9)	**9.3 (4.3)**	46.8 (5.3)
4	7.1 (4.7)	**5.2 (3.5)**	7.5 (6.2)	**69.7 (4.6)**	76.2 (4.1)	**14.9 (3.6)**	72.5 (6.0)
				Sonar			
1	25.0 (11.9)	14.5 (6.4)	**11.0 (7.5)**	50.6 (8.5)	**44.3 (8.7)**	**35.2 (10.6)**	45.2 (8.1)
2	25.0 (11.9)	14.5 (6.4)	**11.0 (7.5)**	**41.3 (6.3)**	53.8 (7.0)	**30.3 (11.6)**	45.2 (9.2)
				Imox			
1	8.8 (4.8)	**4.1 (4.7)**	4.6 (4.5)	**17.6 (8.4)**	74.5 (10.6)	**13.4 (8.6)**	48.8 (8.6)
2	4.7 (5.7)	**0.5 (1.7)**	1.6 (2.5)	**6.8 (7.5)**	66.7 (7.9)	**8.3 (5.1)**	22.4 (6.1)
3	6.3 (4.8)	**0.5 (1.7)**	6.2 (6.8)	**4.2 (4.8)**	17.7 (8.5)	6.7 (5.9)	**4.2 (3.3)**
4	11.4 (7.1)	**4.1 (4.7)**	10.4 (5.0)	**18.8 (9.0)**	49.1 (15.2)	**13.1 (10.9)**	40.2 (12.8)

somewhat better than without example outliers, as should be expected, but in some cases it requires careful optimization of the C_1, C_2 parameters (for instance for the imox dataset which has much overlap between the classes). In most cases the data descriptions with the polynomial kernel perform worse than with the Gaussian kernel, except for a few cases. Note that when the polynomial kernel is used, the data is preprocessed to have zero mean and unit variance along each of the feature directions. When this preprocessing is not applied, the classification performance becomes extremely poor (worse than random guessing).

4. Experiments

Finally in this section we investigate how the SVDD works in a real one-class classification problem. We focus on a machine diagnostics problem: the characterization of a submersible water pump (Tax & Duin, 1999). Here the normal operation of the pump has to be characterized, to distinguish it from all other faulty operation conditions. Both target objects (measurements on a normal operating pump) and negative examples (measurements on a damaged pump) are available. In a test bed several normal and outlier situations can be simulated. The normal situations consist of working conditions with different loads and speeds of the pump. The outlier data contain pumping situations with loose foundation, imbalance and a failure in the outer race of the uppermost ball bearing.

Table 3. The training and testing sample sizes for the different vibration datasets with their settings. A speed setting of 46, 50, 54 Hz and a load setting of 29, 33 (0.1 kW) means that all measurements with these settings are used (i.e. 6 situations in total).

Dataset number	Settings		Training set target obj.	Testing set	
	Speeds (Hz)	Loads (0.1 kW)		Target obj.	Outlier obj.
1	46, 48, 50, 52, 54	25, 29, 33	376	374	1126
2	46, 50, 54	29, 33	137	374	1126
3	46, 54	25, 33	100	391	1109
4	50	25, 29, 33	71	405	1095
5	46, 48, 50, 52, 54	29	133	360	1140

To characterize the normal operation condition of the pump, the operator varies the speed (with possible values of 46, 48, 50, 52, 54 Hz) and the load (25, 29, 33 $\times$ 0.1 kW) of the pump. For each setting, the pump is run for some time. On the pump vibration sensors are mounted. From the recorded time series subsamples are taken and the power spectrum is calculated (containing 64 bins, resulting in a 64 dimensional dataset). By this procedure the normal operation conditions can be simulated and recorded. Thus the target area in the feature space can be indicated. It is not clear, though, what the typical working conditions will be in practice. The pump should be operating somewhere in the target area, but it will probably not uniformly over the whole area.

These characteristics are simulated using this dataset by sampling the training data in different ways. In this way the personal choices of the operator vary, although they should in all cases define the normal operation area in the feature space. In Table 3 five different datasets are listed with their speed and load settings, and the number of training and testing objects available. The first dataset is considered to be the most complete, that it covers the whole working area of the pump. The next datasets are approximations to this dataset.

To see how well the SVDD and the SVDD with negative examples (SVDD_neg) perform, we compare them with a number of other methods. The first method is a normal density where just the mean and the covariance matrix of the data has to be estimated. This will result in a ellipsoidal boundary around the data. In high dimensional feature spaces the covariance matrix of the target set can become singular. The covariance matrix is regularized by adding a small diagonal term to the covariance matrix: $\Sigma' = \Sigma + \lambda I$. The regularization parameter is optimized for the problem at hand and is in most cases between 10^{-3} and 10^{-6}. The second method is the Parzen density where the width of the Parzen kernel is estimated using leave-one-out optimization (Duin, 1976). The third method is a Mixture of Gaussians, optimized using EM (Bishop, 1995). The number of clusters is set to 5 beforehand, but varying this value does not improve performance very much.

The last method is a nearest neighbor method, which compares the local density of an object $\mathbf{x}$ with the density of the nearest neighbor in the target set (Tax & Duin, 2000). The distance between the test object $\mathbf{x}$ and its nearest neighbor in the training set $\mathrm{NN}^{\mathrm{tr}}(\mathbf{x})$ is compared with the distance between this nearest neighbor $\mathrm{NN}^{\mathrm{tr}}(\mathbf{x})$ and its nearest neighbor in the training set $\mathrm{NN}^{\mathrm{tr}}(\mathrm{NN}^{\mathrm{tr}}(\mathbf{x}))$. When the first distance is much larger than the second distance, the object will be regarded as an outlier. We use the quotient between the first and

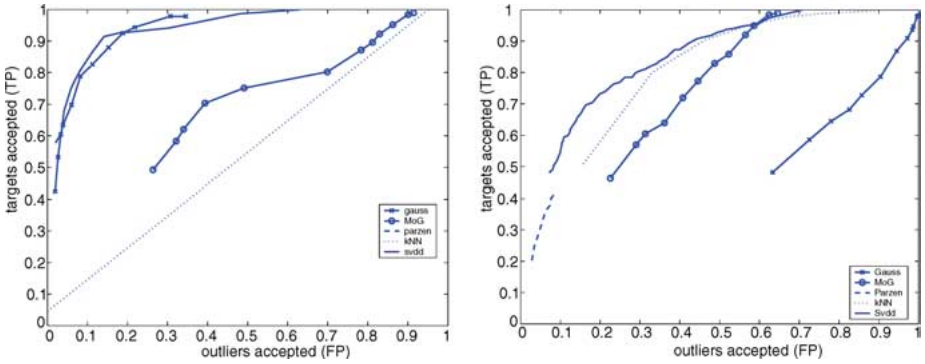

Figure 5. The ROC curves for dataset 1 (see Table 3). Left for the data in the original 64D feature space, right the data in 15D (retaining 80% of the variance).

the second distance as indication of the validity of the object:

$$\rho(\mathbf{x}) = \frac{\|\mathbf{x} - \mathrm{NN}^{\mathrm{tr}}(\mathbf{x})\|}{\|\mathrm{NN}^{\mathrm{tr}}(\mathbf{x}) - \mathrm{NN}^{\mathrm{tr}}(\mathrm{NN}^{\mathrm{tr}}(\mathbf{x}))\|} \tag{37}$$

where $\mathrm{NN}^{\mathrm{tr}}\mathbf{x}$ is the nearest neighbor of $\mathbf{x}$ in the training set.

It is to be expected that when just the target class area is covered, and not sampled according to what will happen in practice, the density methods will fail and the methods which just focus on modeling the data will still work. In the case of a good distribution representation in the training set, the density and the boundary methods should perform about equal.

For the comparison, the trade-off between the fraction of the target class rejected (error of the first kind, $\mathcal{E}_{\mathrm{I}}$) versus the fraction of the outlier accepted (error of the second kind, $\mathcal{E}_{\mathrm{II}}$) is investigated. The methods will be compared over a range of threshold values, from 1% target class rejection up to 50% rejection. The fraction of outliers rejected and target objects accepted is measured afterwards on the test set.

In figure 5 the Receiver-Operating Characteristic curves are shown (ROC-curve) (Metz, 1978) for the methods trained on the power spectrum features. In the left plot the results on the full 64 dimensional feature space is shown. A good discrimination between target and outlier objects means both a small fraction of outlier accepted and a large fraction of target objects accepted. The optimal performing classifier will be in the upper left corner of the ROC curve. The ROC curves in the left plot show, that for this data overlap between the classes occur, and this optimal performance cannot be obtained.

The sample size is too low to obtain a good density estimate and the Parzen density estimator is not able to estimate the density well. It rejects all target and all outlier data. The Gaussian density and the SVDD obtain about the same solution, indicating that in this representation with these amounts of data, no more than an ellipse can be fitted. The Mixture of Gaussians and the kNN overfit and show poor performance.

In the right subplot the data dimensionality is reduced by PCA to retain 80% of the variance. In this case, the sample size is sufficient to follow the boundaries of the target set more closely. The poor performance of the Gaussian shows, that the target distribution

Table 4. Integrated errors for the five outlier detection methods on the original power spectrum data and versions with only the first few principal components.

	Number of features					
Method	3	5	10	20	30	64
Normal dens.	38.1	37.8	34.1	25.9	16.6	**4.5**
Mix.o.Gauss.	18.5	21.0	**9.8**	11.4	14.4	15.2
Parzen dens.	20.8	45.0	45.0	45.0	45.0	45.0
kNN	16.5	13.8	12.1	19.9	22.5	30.4
SVDD	**15.5**	**14.3**	11.6	**10.9**	**9.9**	4.9

Best performances from the methods are shown in bold.

boundary is more complex than a ellipsoid. The Parzen density estimator has still some problems with the sample size, but both the SVDD and the kNN give nice results.

To make a more quantitative comparison, an error measure is derived from the ROC-curves. The fraction of the outliers which is accepted ($\mathcal{E}_{\mathrm{II}}$), is averaged over varying $\mathcal{E}_{\mathrm{I}}$ (Bradley, 1997):

$$\mathcal{E}_M = \int_A^B \mathcal{E}_{\mathrm{II}}(\mathcal{E}_{\mathrm{I}})\, d\mathcal{E}_{\mathrm{I}} \tag{38}$$

In Table 4 the results on the power spectrum data are shown. The methods are applied to both the original 64 dimensional data set and to the first principal components of that data, ranging from 3 up to 30 features. For higher dimensionalities the normal densities and the SVDD perform well. This indicates that for higher dimensions, approximating the target class by one ellipse is sufficient. For the Normal density, the Mixture of Gaussians and the SVDD the error decreases with increasing dimensionality while for the Parzen estimator it decreases. On low dimensionalities the density can be estimated relatively well by the Parzen estimator, but it suffers from the fact that it is not capable of accepting all target objects in the test set. This means that in order to compute the AUC error from the ROC curve, the ROC-curve should be extended. Here it is chosen to extend it in the worst case sense (when a certain fraction target acceptance cannot be reached, it is assumed that the fraction of outliers accepted will be 1). Therefore the errors for the Parzen density estimator are high.

The use of the negative examples in the SVDD is problematic in this example, because there are many outliers and some show complete overlap with the target class. Choosing a set of 'good' outliers improves performance significantly, but when some 'poor' outliers are chosen (outliers which are actually inside the target class) performance may deteriorate to random guessing. In these cases a manual optimization of C_1 and C_2 is required, by judging what fraction of the outliers are genuine outliers and which are actually not. When randomly outliers are drawn, the variance of the different outcomes is too large to be useful.

In figures 6 and 7 the ROC curves for the one-class classifiers are given for the other datasets listed in Table 3. By the (sub)sampling of the training set, the distribution of the training target data does not completely reflect the target distribution in the test data, although it should define the same area in feature space. In almost all cases the SVDD,

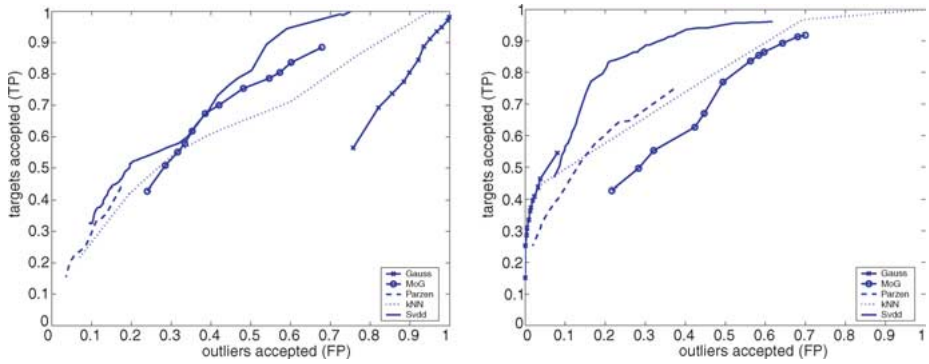

Figure 6. The ROC curves for dataset 2 and 3 (see Table 3). The data dimensionality was reduced by PCA to retain 90% of the variance.

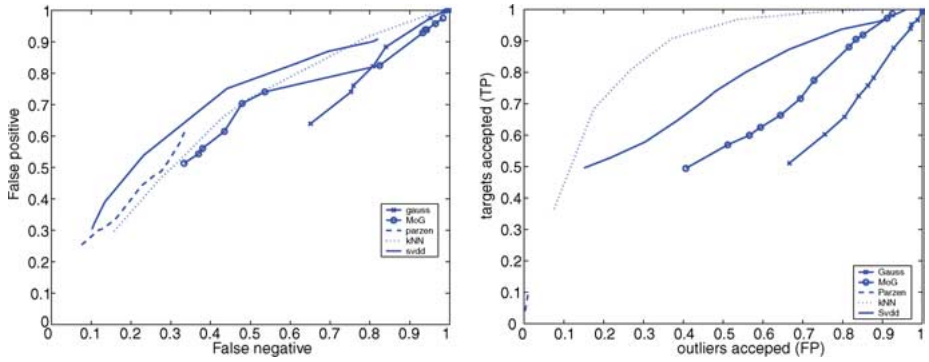

Figure 7. The ROC curves for datasets 4 (on the left) and 5 on the right). (see Table 3). The data dimensionality was reduced by PCA to retain 90% of the variance.

which focuses on modeling the boundary, obtains a better performance than the density estimates (this is very clear in datasets 3 and 5). The performance of the density estimators can be improved with respect to the SVDD by reducing the dimensionality, but in these cases the absolute performance is so poor, that it is not useful to apply in practice.

The worst classification performances are obtained for dataset 4, in which measurements on just one speed (50 Hz.) was included. Using just one speed setting is insufficient to define the whole feature area, in contrast to dataset 5, where a single load is used, but where the performance of the SVDD is still very good. These examples show that, when an area in feature space should be characterized (and not per se the complete target probability density), the SVDD gives very good data descriptions. The density estimators on the other hand, focus much more on the high density areas, which are not representative for the practical situation.

5. Conclusions

In this paper we discussed a new method to solve the multidimensional outlier detection problem. Instead of estimating a probability density, it obtains a boundary around the data

set. By avoiding the estimation of the data density, it can obtain a better data boundary. The data description is inspired by the Support Vector Classifier. The boundary is described by a few training objects, the support vectors. It is possible to replace normal inner products by kernel functions and thus to obtain more flexible data descriptions. In contrast to the Support Vector Classifier, the Support Vector Data Description using a polynomial kernel suffers from the large influence of the norms of the object vectors, but it shows promising results for the Gaussian kernel. Using the Gaussian kernel, descriptions comparable to the hyperplane solution by Schölkopf et al. are obtained.

The fraction of the target objects which become support vectors is an estimate of the fraction of target objects rejected by the description. When the maximum desired error on the target set is known beforehand, the width parameter s can be set to give the desired number of support vectors. When not enough objects are available, the fraction of support vectors stays high whatever width parameter s is used. This is an indication that more data is necessary. Extra data in the form of outlier objects can also be used to improve the Support Vector Data Description.

Comparing the Support Vector Data Description with other outlier detection methods, Normal density estimation, Parzen density estimation and the Nearest Neighbor method, it shows comparable or better results for sparse and complex data sets, especially when outlier information is used. For very small target error rates the SVDD breaks down, while for high sample sizes a density estimation like Parzen density method is to be preferred.

Acknowledgments

This work was partly supported by the Foundation for Applied Sciences (STW) and the Dutch Organization for Scientific Research (NWO).

Note

1. Or actually 2, but this requires a very specific placement of the vectors, on opposite sides with equal distance to the center.

References

Barnett, V., & Lewis, T. (1978). *Outliers in Statistical Data*, 2nd ed. Wiley series in probability and mathematical statistics. John Wiley & Sons Ltd.

Bishop, C. (1994). Novelty detection and neural network validation. *IEE Proceedings on Vision, Image and Signal Processing. Special Issue on Applications of Neural Networks, 141:4*, 217–222.

Bishop, C. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press, Walton Street, Oxford OX2 6DP.

Blake, C., Keogh, E., & Merz, C. (1998). UCI repository of machine learning databases. http://www.ics.uci.edu/∼mlearn/MLRepository.html, University of California, Irvine, Dept. of Information and Computer Sciences.

Bradley, A. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition, 30:7*, 1145–1159.

Duin, R. (1976). On the choice of the smoothing parameters for Parzen estimators of probability density functions. *IEEE Transactions on Computers, C-25:11*, 1175–1179.

Japkowicz, N., Myers, C., & Gluck, M. (1995). A novelty detection approach to classification. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* (pp. 518–523).

Koch, M., Moya, M., Hostetler, L., & Fogler, R. (1995). Cueing, feature discovery and one-class learning for synthetic aperture radar automatic target recognition. *Neural Networks, 8:7/8*, 1081–1102.

MacKay, D. (1992). Bayesian methods for adaptive models. Master's thesis, California Institute of Technology, Pasadena, California.

Metz, C. (1978). Basic principles of ROC analysis. *Seminars in Nuclear Medicine, VIII:4*.

Moya, M., & Hush, D. (1996). Network contraints and multi-objective optimization for one-class classification. *Neural Networks, 9:3*, 463–474.

Moya, M., Koch, M., & Hostetler, L. (1993). One-class classifier networks for target recognition applications. In *Proceedings World Congress on Neural Networks* (pp. 797–801). Portland, OR: International Neural Network Society.

Parra, L., Deco, G., & Miesbach, S. (1996). Statistical independence and novelty detection with information preserving nonlinear maps. *Neural Computation, 8*, 260–269.

Richard, M., & Lippmann, R. (1991). Neural network classifiers estimate Bayesian a posteriori probabilities. *Neural Computation, 3*, 461–483.

Ripley, B. (1996). *Pattern Recognition and Neural Networks*. Cambridge: Cambridge University Press.

Ritter, G., & Gallegos, M. (1997). Outliers in statistical pattern recognition and an application to automatic chromosome classification. *Pattern Recognition Letters, 18*, 525–539.

Roberts, S., Tarassenko, L., Pardey, J., & Siegwart, D. (1994). A validation index for artificial neural networks. In *Proceedings of Int. Conference on Neural Networks and Expert Systems in Medicine and Healthcare* (pp. 23–30).

Roberts, S., & Penny, W. (1996). Novelty, confidence and errors in connectionist systems. Tech. rep., Imperial College, London. TR-96-1.

Rosen, J. (1965). Pattern seperation by convex programming. *Journal of Mathematical Analysis and Applications, 10:1*, 123–134.

Schölkopf, B. (1997). Support vector learning. Ph.D. thesis, Technischen Universität Berlin.

Schölkopf, B., Burges, C., & Vapnik, V. (1995). Extracting support data for a given task. In U. Fayyad, & R. Uthurusamy (eds.), *Proc. of First International Conference on Knowledge Discovery and Data Mining* (pp. 252–257). Menlo Park, CA. AAAI Press.

Schölkopf, B., Platt, J., Shawe-Taylor, J., A., S., & Williamson, R. (1999a). Estimating the support of a high-dimensional distribution. *Neural Computation, 13:7*.

Schölkopf, B., Williamson, R., Smola, A., & Shawe-Taylor, J. (1999b). SV estimation of a distribution's support. In *Advances in Neural Information Processing Systems*.

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

Tarassenko, L., Hayton, P., & Brady, M. (1995). Novelty detection for the identification of masses in mammograms. In *Proc. of the Fourth International IEE Conference on Artificial Neural Networks* Vol. 409 (pp. 442–447).

Tax, D., Ypma, A., & Duin, R. (1999). Support vector data description applied to machine vibration analysis. In M. Boassen, J. Kaandorp, J. Tonino, & V. M.G. (eds.), *Proceedings of the Fifth Annual Conference of the ASCI* (pp. 398–405).

Tax, D., & Duin, R. (1999). Support vector domain description. *Pattern Recognition Letters, 20:11–13*, 1191–1199.

Tax, D., & Duin, R. (2000). Data descriptions in subspaces. In *Proceedings of the International Conference on Pattern Recognition 2000*, Vol. 2 (pp. 672–675).

Vapnik, V. (1998). *Statistical Learning Theory*. Wiley.

Received December 13, 2000
Revised May 27, 2002
Accepted June 10, 2002
Final manuscript June 17, 2002