World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:7, No:7, 2013

Autonomously Determining the Parameters for SVDD with RBF Kernel from a One-Class Training Set

Andreas Theissler, Ian Dear

International Science Index, Computer and Information Engineering Vol:7, No:7, 2013 waset.org/Publication/16382

Abstract—The one-class support vector machine "support vector data description" (SVDD) is an ideal approach for anomaly or outlier detection. However, for the applicability of SVDD in real-world applications, the ease of use is crucial. The results of SVDD are massively determined by the choice of the regularisation parameter C and the kernel parameter σ of the widely used RBF kernel. While for two-class SVMs the parameters can be tuned using cross-validation based on the confusion matrix, for a one-class SVM this is not possible, because only true positives and false negatives can occur during training. This paper proposes an approach to find the optimal set of parameters for SVDD solely based on a training set from one class and without any user parameterisation. Results on artificial and real data sets are presented, underpinning the usefulness of the approach.

Keywords—Support vector data description, anomaly detection, one-class classification, parameter tuning.

I. Introduction

Anomaly detection [1] has gained importance over the last years. Recording data from technical systems is typically no challenge nowadays, data acquisition systems have become compact and cheap. Among the applications are system health monitoring, the analysis of log files from computer systems, intrusion detection, performance monitoring, or the identification of potential errors in data measured from electronic or mechanical systems. The authors of this paper conduct research on the detection of potential errors in recordings from test drives [2], [3].

An anomaly is a deviation from expected behaviour [4], other terms are novelty and outlier [5]. The detection of anomalies can be automated by training an anomaly detection system on a labelled training set with normal and abnormal data and have the system classify unseen data. This corresponds to a two-class classification problem. The task is to assign an unclassified instance to either the normal class ω_n or the abnormal class ω_a based on a set of features f.

By using normal and abnormal training data sets, the decision boundary is heavily influenced by the choice of the abnormal data. Using a non-representative training data set of anomalies, an incorrect decision function is learned. For example using anomaly detection to detect faults, it is very unlikely to have a representative data set. In some cases there are no abnormal data sets available. On the other hand the amount of training data containing normal instances is not constrained, since the training data can be obtained by recording data from a system in normal operation mode.

To overcome the mentioned limitations of two class-classification approaches, an alternative for anomaly detection systems is to only learn the normal behaviour and classify deviations as abnormal. In other words, the training period is exclusively conducted on a training set of normal instances.

Support vector machines (SVM) have shown to yield good results on classification tasks and have widely been used. While two-class support vector machines separate the classes by a hyperplane [6], [7], in [8] the one-class SVM "support vector data description" (SVDD) was introduced to cope with the problem of one-class classification. SVDD finds a closed decision boundary, a hypersphere, around the normal instances in the training data set using a so-called kernel function. It is therefore ideal for anomaly detection.

A. Motivation

For the applicability of a detection algorithm in real-world applications, the ease of use is crucial. Users are domain-experts like physicists, system administrators, or test engineers. They should not have to be concerned with the complexity of machine learning algorithms and their parameterisation.

Unfortunately, the results of SVDD are massively determined by the choice of the regularisation parameter C and the kernel parameter used during training. A bad choice of those parameters makes the results useless.

In two-class SVMs, the parameters are tuned using cross-validation or leave-one-out validation based on a labelled training data set containing both classes. For a given set of parameters, true positives, false positives, true negatives, and false negatives are measured. Based on these values the parameters are optimised w.r.t. for example the overall-accuracy, the true positive rate, or further metrics [9] depending on the application.

Since in one-class classification problems only instances from the normal class are contained in the training data set, only true positives and false negatives can be measured, where positive refers to the normal class ω_n in this paper. As a consequence, parameter tuning cannot be done as for two-class SVMs.

This paper addresses the question, if it is possible to find a set of parameters for SVDD, that yields good results

1) solely on the training set of normal instances
2) without user parameterisation

Andreas Theissler is with IT-Designers, Germany and Brunel University, West London, UK

Ian Dear is with Brunel University, West London, UK

II. Related work

Few publications have addressed the problem of finding the parameters for SVDD. If there is certainty, that no anomalies exist in the training data set, C can be set to 1 [10], so that the hypersphere will include all instances. If the fraction of outliers ν in the training data set is known, the parameter should be set to $C \leq \frac{1}{\nu N}$ [10].

In [11], the authors discuss optimising parameters for SVDD in the context of text classification. SVDD is used to learn from the normal class and then optimised using instances from the outlier class, i.e. the problem is transformed to two-class classification.

In the absence of outlier data, [12] proposes to generate artificial instances that are uniformly distributed in a hypersphere around the normal class including the region of the normal class. Based on the fraction of instances classified as normal an optimisation criterion is defined.

In [13] an error function for SVDD with the RBF kernel is defined utilising the number of support vectors, that is used to optimise the two parameters without the need to select or generate outliers. However, a trade-off parameter is introduced and the results are reported to be rather weak.

In [10] an estimation of the error on the normal class is given, based on the number of essential support vectors. Identifying the essential support vectors is done using leave-one-out. By leaving out an instance within the boundary or a non-essential support vector from the training set, the decision boundary is unchanged. On the other hand, leaving out an essential support vector, the decision boundary covers a smaller region. However, leave-one-out is not practical for large training sets due to its high execution time, e.g. for 10000 instances, 10000 training runs are required.

III. Support vector data description

Support vector data description (SVDD) was introduced in [8] as a one-class SVM. SVDD finds a hypersphere around the normal instances in the training data set. The hypersphere is fully determined by its radius R and its center $\vec{a}$, as illustrated in Fig. 1, and is found by solving the optimisation problem of minimising:

1) the error on the normal class, i.e. false negatives
2) the chance of misclassifying data from the abnormal class, i.e. false positives

Minimising the error on the normal class is achieved by adjusting R and $\vec{a}$ in a way that all instances of the training data set are contained in the hypersphere. On the other hand, minimising the chance of misclassifying data from the abnormal class cannot be achieved straightforward, since in the absence of abnormal training data, false positives cannot be measured during the optimisation step.

A. Finding the optimal hypersphere

A hypersphere with an infinite volume would obviously enclose all instances but misclassify all abnormal instances. So the hypersphere's volume is used as a second optimisation criterion. The trade-off between the number of misclassified normal instances and the volume of the normal region is optimised. On one hand the decision boundary is desired to capture the normal instances, while on the other hand keeping the hypersphere's volume at a minimum. Hence, the following optimisation problem is to be solved [14]:

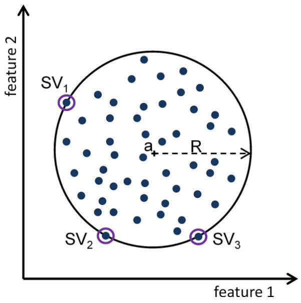


Fig. 1. A hypersphere in a 2-dimensional feature space with radius R and center a is described by the three support vectors $SV_1 \cdots SV_3$.

minimise

$$F(R, \vec{a}) = R^2 \qquad (1)$$

subject to

$$\|\vec{x_i} - \vec{a}\|^2 \leq R^2 \quad \forall i \qquad i = 1, .., M \qquad (2)$$

where $\vec{x_i}$ denotes the instances and M the number of instances in the training data set, $\vec{a}$ is the hypersphere's center, and $\|\vec{x_i} - \vec{a}\|$ is the distance between $\vec{x_i}$ and $\vec{a}$.

The distance $\|\vec{x_i} - \vec{a}\|$ is calculated by $\sqrt{(\vec{x_i} - \vec{a}) \cdot (\vec{x_i} - \vec{a})}$. Since calculating the square root is computationally expensive, the squared distance $\|\vec{x_i} - \vec{a}\|^2$ is used and compared to the squared radius R^2. The squared distance can be reformulated using the binomial theorem which is beneficial, as will become clear in Section III-D.

The center $\vec{a}$ is implicitly described by a linear combination of selected instances from the training data set, the so-called support vectors. The remaining instances are discarded.

Having solved the optimisation problem eq. (1) and eq. (2) and hence having found the hypersphere, for each $\vec{x_i}$ one of two terms is satisfied. This is used to select those $\vec{x_i}$ that become support vectors:

$$\|\vec{x_i} - \vec{a}\|^2 < R^2 \quad \rightarrow \quad \vec{x_i} \text{ is within the hypersphere} \qquad (3)$$

$$\|\vec{x_i} - \vec{a}\|^2 = R^2 \quad \rightarrow \quad \vec{x_i} \text{ is used as a support vector} \qquad (4)$$

As an illustrative example one could think of a 2-dimensional training data set with 50 instances. In this case the decision boundary would be a circle, which is fully described by three distinct points on its circumference. Hence, ideally three support vectors would be selected and 47 instances be discarded as illustrated in Fig. 1.

Classifying a test instance $\vec{x_t}$ is a matter of determining whether it is inside or outside the hypersphere, which is done by solving $\|\vec{x_t} - \vec{a}\|^2 < R^2$.

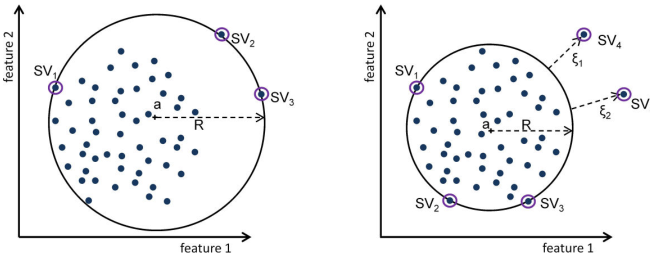


Fig. 2. The introduction of the slack variables ξ_i allows for some instances of the training data set to be outside the decision boundary.

B. Reducing the sensitivity to outliers

Demanding that *all* instances are contained in the hypersphere means that outliers contained in the training data set will massively influence the decision boundary. So SVDD in this form is very sensitive to outliers, which is not desired.

Analogous to hard-margin SVMs, that can be transformed to soft-margin SVMs by allowing some instances to be on the wrong side of the separating hyperplane [7], in SVDD slack variables are introduced. These slack variables ξ_i allow for some instances $\vec{x_i}$ in the training data set to be outside the hypersphere as shown in Fig. 2.

The parameter C is introduced controlling the influence of the slack variables and thereby the error on the normal class and the hypersphere's volume. So the optimisation problem of eq. (1) and eq. (2) changes to minimising [14]:

$$F(R, \vec{a}, \xi_i) = R^2 + C \sum_{i=1}^{M} \xi_i \tag{5}$$

subject to

$$\|\vec{x_i} - \vec{a}\|^2 \leq R^2 + \xi_i \quad \forall i \tag{6}$$

and

$$\xi_i \geq 0 \quad \forall i \tag{7}$$

C. Solving the optimisation problem

As described in [10], the optimisation problem is solved by incorporating the constraints eq. (6) and eq. (7) into eq. (5) using the method of Lagrange for positive inequality constraints [15]. This allows to transform a constrained optimisation problem into an unconstrained one by integrating the constraints into the equation to be optimised. First eq. (6) is rewritten to become a positive inequality constraint:

$$R^2 + \xi_i - \|\vec{x_i} - \vec{a}\|^2 \geq 0 \tag{8}$$

For a function f and two constraints $g_1 \geq b_1$ and $g_2 \geq b_2$, the Lagrangian is formulated as $L = f - \alpha(g_1 - b_1) - \beta(g_2 - b_2)$, introducing the so-called Lagrange multipliers α_i and β_i. Incorporating constraints eq. (7) and eq. (8) into eq. (5), the optimisation problem changes into maximising

$$\begin{aligned} L(R, \vec{a}, \alpha_i, \beta_i, \xi_i) &= R^2 + C \sum_{i=1}^{M} \xi_i \qquad (9) \\ &- \sum_{i=1}^{M} \alpha_i (R^2 + \xi_i - x_i^2 + 2\vec{a} \cdot \vec{x_i} - \vec{a}^2) - \sum_{i=1}^{M} \beta_i \xi_i \end{aligned}$$

The partial derivatives are set to 0, which for R is

$$\frac{\partial L}{\partial R} = 2R - 2R \sum_{i=1}^{M} \alpha_i \overset{!}{=} 0 \tag{10}$$

and yields the condition

$$\sum_{i=1}^{M} \alpha_i = 1 \tag{11}$$

The partial derivative with respect to $\vec{a}$

$$\frac{\partial L}{\partial \vec{a}} = -2 \sum_{i=1}^{M} (\alpha_i \vec{x_i} - \alpha_i \vec{a}) \overset{!}{=} 0 \tag{12}$$

can be reformulated, the -2 be dropped, and then equals 0, if

$$\vec{a} = \frac{\sum_{i=1}^{M} \alpha_i \vec{x_i}}{\sum_{i=1}^{M} \alpha_i} = \sum_{i=1}^{M} \alpha_i \vec{x_i} \quad \text{with} \ \sum_{i=1}^{M} \alpha_i = 1 \quad \text{from eq. (11)}$$

which shows that the center $\vec{a}$ is expressed as a linear combination of the support vectors. Finally, deriving with respect to ξ_i leads to

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - \beta_i \overset{!}{=} 0 \tag{13}$$

Since $\alpha_i \geq 0$ and $\beta_i \geq 0$, and $\beta_i = C - \alpha_i$ this allows to drop β_i by instead adding the following constraint

$$0 \leq \alpha_i \leq C \tag{14}$$

Resubstituting the found constraints yields a less complex equation. First eq. (9) is reformulated as

$$\begin{aligned} L =& R^2 + C \sum_{i=1}^{M} \xi_i - \sum_{i=1}^{M} \alpha_i R^2 - \sum_{i=1}^{M} \alpha_i \xi_i + \sum_{i=1}^{M} \alpha_i \vec{x_i}^2 \\ &- 2 \sum_{i=1}^{M} \alpha_i \vec{a} \cdot \vec{x_i} + \sum_{i=1}^{M} \alpha_i \vec{a}^2 - \sum_{i=1}^{M} \beta_i \xi_i \end{aligned} \tag{15}$$

Substituting $(\sum_{i=1}^{M} \alpha_i) = 1$ from eq. (11), $\beta_i = C - \alpha_i$ from eq. (14), and $\vec{a} = (\sum_{i=1}^{M} \alpha_i \vec{x_i})$ from eq. (12), the optimisation problem changes into maximising

$$L(\alpha) = \sum_{i=1}^{M} \alpha_i (\vec{x_i} \cdot \vec{x_i}) - \sum_{i,j=1}^{M} \alpha_i \alpha_j (\vec{x_i} \cdot \vec{x_j}) \tag{16}$$

subject to

$$0 \leq \alpha_i \leq C \quad \forall i \tag{17}$$

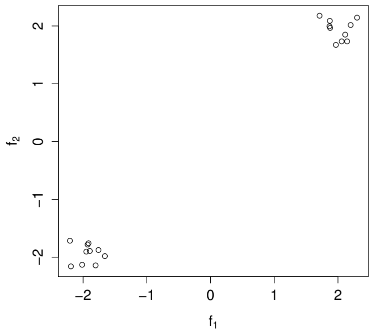


Fig. 3. Instances from the normal class distributed in two clusters. Enclosing all instances with a sphere would massively overestimate the normal class.

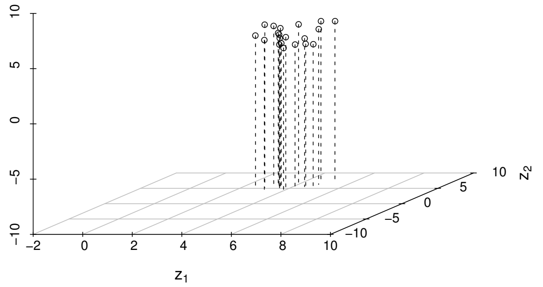


Fig. 4. The instances of the contrived data set can be enclosed by a sphere in the three-dimensional feature space created by the mapping function.

Having determined all α_i, the parameters $\vec{a}$ and ξ_i can be deduced. The radius R is determined by picking an arbitrary support vector $\vec{x_i}$ on the boundary, i.e. with $0 < \alpha_i < C$, and solving $R = \|\vec{x_i} - \vec{a}\|$.

D. Introducing non-spherical decision boundaries

At this point, SVDD is capable of surrounding the normal data by a hypersphere. It is rare that in classification problems the distribution of the data is spherical. Hence, SVDD in this form would yield poor classification results for most data sets. SVDD maps the data to higher-dimensional space, where it can be surrounded by a hypersphere.

An example of such a mapping is given for the contrived two-dimensional data set in Fig. 3. The data has instances from the normal class distributed in two clusters. Enclosing all instances with a circle would massively overestimate the normal class. Using an example mapping $Z := (f_1^2, \sqrt{2} f_1 f_2, f_2^2)$ [6], the data set can be mapped to a three dimensional space, as shown in Fig. 4, where the instances can be surrounded by a sphere. This mapping is ideal for the contrived data set, but it is not practical for arbitrary data sets, if the ideal mapping has to be known.

As can be seen from eq. (16) $\vec{x_i}$ and $\vec{x_j}$ are solely incorporated as the inner products (scalar products) $(\vec{x_i} \cdot \vec{x_i})$ and $(\vec{x_i} \cdot \vec{x_j})$ respectively. Instead of actually mapping each instance to a higher-dimensional space using a mapping function $\phi()$, the so-called kernel trick is used to replace the inner products $(\phi(\vec{x_i}) \cdot \phi(\vec{x_j}))$ by a kernel function $K(\vec{x_i}, \vec{x_j})$. The mapping is implicitly done by solving $K(\vec{x_i}, \vec{x_j})$.

$$K(\vec{x_i}, \vec{x_j}) = \phi(\vec{x_i}) \cdot \phi(\vec{x_j}) \tag{18}$$

So eq. (16) becomes:

$$L = \sum_{i=1}^{M} \alpha_i K(\vec{x_i}, \vec{x_i}) - \sum_{i,j=1}^{M} \alpha_i \alpha_j K(\vec{x_i}, \vec{x_j}) \tag{19}$$

A variety of kernel functions have been proposed. Two widely used kernels are the polynomial kernel and the radial basis function (RBF) kernel, also referred to as the Gaussian kernel.

E. Classifying a test instance

A test instance $\vec{x_t}$ is classified by solving $\|\vec{x_t} - \vec{a}\|^2 \overset{?}{>} R^2$. The squared distance $\|\vec{x_t} - \vec{a}\|^2$ can be rewritten as $(\vec{x_t} \cdot \vec{x_t} - 2\vec{x_t} \cdot \vec{a} + \vec{a} \cdot \vec{a})$. Replacing $\vec{a}$ by its linear combination of support vectors from eq. (13) yields

$$\vec{x_t} \cdot \vec{x_t} - 2 \sum_{i=1}^{M} \alpha_i (\vec{x_t} \cdot \vec{x_i}) + \sum_{i,j=1}^{M} \alpha_i \alpha_j (\vec{x_i} \cdot \vec{x_j}) \tag{20}$$

Again, the inner products are replaced by the kernel function used during training. A test instance is classified as abnormal, if the following inequality holds.

$$\begin{aligned} R^2 < & K(\vec{x_t}, \vec{x_t}) - 2 \sum_{i=1}^{M} \alpha_i K(\vec{x_t}, \vec{x_i}) \\ & + \sum_{i,j=1}^{M} \alpha_i \alpha_j K(\vec{x_i}, \vec{x_j}) \end{aligned} \tag{21}$$

F. The RBF kernel

The RBF kernel is reported to be most suitable to be used with SVDD in [10]. As opposed to the polynomial kernel, the RBF kernel does not depend on the position of instances with respect to the origin [10]. This kernel adds only one parameter to the classification problem, the kernel width σ. The RBF kernel is given by

$$K(\vec{x_i}, \vec{x_j}) = e^{-\frac{\|\vec{x_i} - \vec{x_j}\|^2}{\sigma^2}} \tag{22}$$

The kernel function can take on values from the interval $(0, 1]$ and converges to 0 for high distances $\|\vec{x_i} - \vec{x_j}\|$. Since $K(\vec{x_i}, \vec{x_i}) = e^{-\frac{\|\vec{x_i} - \vec{x_i}\|^2}{\sigma^2}} = 1$ and $\sum_{i=1}^{M} \alpha_i = 1$, eq. (16) can be simplified for the RBF kernel:

$$L(\alpha) = 1 - \sum_{i,j=1}^{M} \alpha_i \alpha_j K(\vec{x_i} \cdot \vec{x_j}) \tag{23}$$

subject to

$$0 \leq \alpha_i \leq C \quad \forall i \tag{24}$$

The equation to classify an instance eq. (21) boils down to:

$$1 - 2\sum_{i=1}^{M} \alpha_i e^{-\frac{\|\vec{x_t} - \vec{x_i}\|^2}{\sigma^2}} + \sum_{i,j=1}^{M} \alpha_i \alpha_j e^{-\frac{\|\vec{x_i} - \vec{x_j}\|^2}{\sigma^2}} \overset{?}{>} R^2 \tag{25}$$

IV. Autonomously tuning the SVDD parameters

Like most classifiers, SVDD has parameters that massively influence the classification accuracy. While SVDD yields good classification results for the one-class problem, manually adjusting the parameters C and σ makes it non-applicable for real-world applications.

As surveyed in Section II, current approaches either work with available or generated outlier data sets or by heuristically or experimentally setting the parameters. The autonomous approach proposed in [13] is reported to yield poor decision boundaries. This section presents an approach to autonomously tune the parameters by solely working on the training set. Using grid search, parameter candidates are selected and based on an optimisation criterion the best pair $\{C_i, \sigma_i\}$ is determined. The task is to tune the parameters so that the accuracy on the test set and on unseen data is optimal. The proposed approach is based on the findings in a bachelor thesis supervised by this paper's main author [16], that a radius close to 1 appears to yield good solutions for the RBF kernel.

It is proposed to scale all input features independently to a value range of $[-1, 1]$ by min/max normalisation.

The optimisation problem eq. (5) is solved for a given set of parameters C and σ. Determining those parameters is not straightforward. The regularisation parameter C, introduced in eq. (5), is lower-bound by $\frac{1}{N}$, where N is the number of instances in the training data set. $C = 1$ corresponds to the hard-margin solution, where all instances are enclosed in the decision boundary [8]. So the value range of C is

$$\frac{1}{N} \leq C \leq 1 \tag{26}$$

The second parameter to be optimised is the kernel width σ. For high values of σ the boundary will become spherical with the risk of underfitting, while for small values of σ a high fraction of instances are selected to be support vectors, hence the boundary is very flexible and is prone to overfitting.

A. Selecting parameter candidates using grid search

The SVDD parameters C and σ are optimised using grid search. Grid search can be considered a brute-force way to optimise parameters, in contrast to e.g. gradient descent. Within a given value range, the grid search algorithm selects candidate values and tests all candidates. The selection of candidate values is done iteratively. While this becomes computationally expensive for many parameters, it is feasible for the two parameters in the problem presented.

The input parameters for the grid search are C_{min}, C_{max}, σ_{min}, σ_{max}, the number of candidates in the current range denoted by τ and some abortion criterion like the number of iterations i.

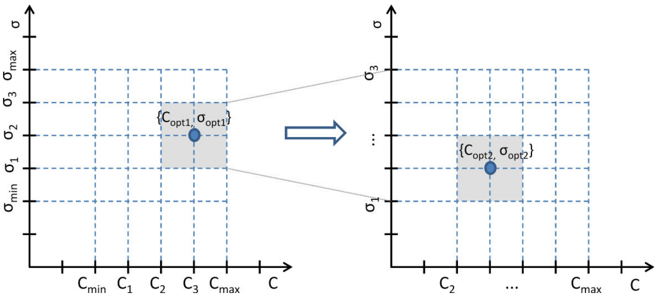


Fig. 5. Functioning of grid search to optimise the SVDD parameter C and σ with $\tau = 5$ and linear partitioning of the ranges. (a) first iteration (b) second iteration

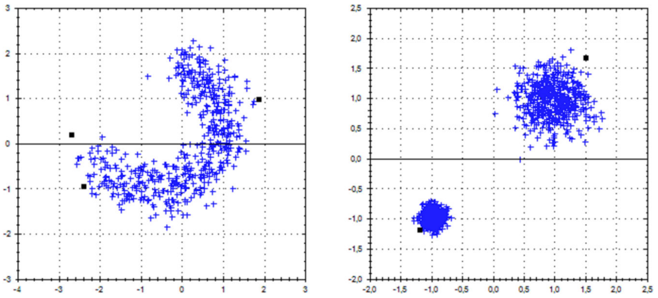


Fig. 6. Results of parameter tuning by minimising the error rate visualised in input feature space. The selected support vectors do not tightly enclose the training set (black squares: support vectors).

For the parameter C, τ values selected within the range of $[C_{min}; C_{max}]$, and for σ respectively. This sums up to τ^2 pairs $\{C_i, \sigma_i\}$ as shown in Fig. 5a. For all $\{C_i, \sigma_i\}$, SVDD is trained and one optimal parameter set $\{C_{opt1}, \sigma_{opt1}\}$ is found.

This is refined in a second iteration by again selecting τ^2 pairs $\{C_i, \sigma_i\}$ in the range of $[C_{opt1-1}; C_{opt1+1}]$ and $[\sigma_{opt1-1}; \sigma_{opt1+1}]$ as shown in Fig. 5b.

This process is repeated until some abortion criterion is met, e.g. until the optimisation criterion does not improve in an iteration or the desired number of iterations is reached.

B. Proposed optimisation criterion

It is crucial to identify a good optimisation criterion. Obviously a low error rate is desired, so selecting $\{C_i, \sigma_i\}$ where the error is minimal seems to be a good approach. However, as can be seen from experimental results, this approach overestimates the region of the normal class by selecting too few support vectors (see Fig. 6). As a consequence the learnt decision boundary does not generalise well.

A solution could be to minimise the error on the normal class, while at the same time maximising the number of support vectors. But, as reported in [14] and confirmed by experiments, the error rate and the number of support vectors are approximately linearly correlated. Hence, optimising the trade-off between the error and the number of support vectors is not possible.

The parameters C and σ are optimal if in the transformed feature space, the instances are arranged in a spherical way.

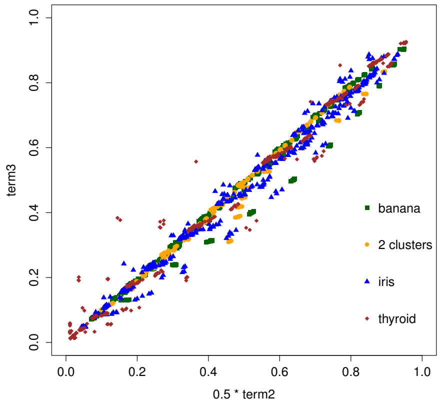


Fig. 7. The values of the second and third term are approximately equal for various data sets over the entire range of the parameters.

Only then will a hypersphere be the ideal decision boundary. So the idea is, to find an optimisation criterion which selects that pair of parameters that best map the data set to one spherical cluster in the transformed feature space.

The hypersphere's radius can be determined by selecting an arbitrary support vector on the boundary $\vec{x_b}$, where $\vec{x_b}$ can be any $\vec{x_i}$ for which $0 < \alpha_i < C$ holds. The radius is then the distance between $\vec{x_b}$ and the center $\vec{a}$, which for the RBF kernel is given by

$$R^2 = 1 - 2\sum_{i=1}^{M} \alpha_i K(\vec{x_b}, \vec{x_i}) + \sum_{i,j=1}^{M} \alpha_i \alpha_j K(\vec{x_i}, \vec{x_j}) \qquad (27)$$

The third term in eq. (27) is exactly the term that is minimised in in the optimisation problem eq. (23). The geometric interpretation in the original space is that $\vec{a} \cdot \vec{a}$ is minimised when $\|\vec{a}\|$ is minimised, which is the case when the center $\vec{a}$ is located at the origin.

The second term $\sum_{i=1}^{M} \alpha_i K(\vec{x_b}, \vec{x_i})$ in eq. (27) incorporates one selected support vector $\vec{x_b}$ on the left side of the kernel function and all support vectors $\vec{x_i}$ on the right side, which is a subset of the third term $\sum_{i,j=1}^{M} \alpha_i \alpha_j K(\vec{x_i}, \vec{x_j})$, where all support vectors are incorporated on both sides of the kernel function.

The second and the third term are proportional. Neglecting the constant 2, they are in fact approximately equal:

$$\sum_{i=1}^{M} \alpha_i K(\vec{x_b}, \vec{x_i}) \approx \sum_{i,j=1}^{M} \alpha_i \alpha_j K(\vec{x_i}, \vec{x_j}) \qquad (28)$$

Experiments have confirmed, that eq. (28) holds for all tested data sets over the entire range of the parameters, as depicted in Fig. 7.

Substituting the terms in eq. (28) by b simplifies eq. (25) into following approximation

$$R^2 \approx 1 - 2b + b \qquad (29)$$

which can be rewritten as

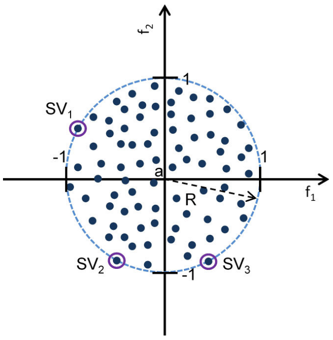


Fig. 8. Optimal mapping in a constructed transformed feature space. The instances are arranged in a spherical way.

$$R \approx \sqrt{1 - b} \qquad (30)$$

From eq. (30) it can be seen that the smaller b is, the closer the radius is to 1. Hence, a radius of 1 can be considered optimal. Fig. 8 shows the optimal solution for the radius in the imaginary mapped feature space: the center is at the origin and the radius is 1.

From the kernel function eq. (22) it can be seen, that the smaller σ is, the closer R will be to 1. So small values of σ are favoured when optimising for $R = 1$. However, for small values of σ, very flexible decision boundaries are obtained, i.e. very many support vectors are selected. This tends to overfit the training set, which in turn yields a high error rate. So only optimising for the radius is insufficient.

In order to find $\{C_{opt}, \sigma_{opt}\}$, the following optimisation criterion is formulated. Informally spoken, the error on the normal class is to be minimised, while at the same time R is desired to be close to 1. Finding R close to 1 is equivalent to minimising $|1 - R|$. Equally weighting error rate and radius this boils down to finding the pair $\{e_i, R_i\}$ closest to the origin by minimising:

$$\lambda_i = \sqrt{e_{\omega_{n_i}}^2 + |1 - R_i|^2} \quad \forall i \qquad (31)$$

This way, non-optimal mappings are assigned larger distances by eq. (31), because non-spherical shapes in the transformed feature space result in a surrounding sphere with $R \neq 1$.

For each tuning step, the error rate and the radius R are determined using k-fold. The training set is randomly split into k folds: $k - 1$ training sets and 1 validation set. The instances in the validation set are classified, the error e_{ω_n} and R are averaged over the k runs.

Fig. 9 shows results based on the proposed optimisation criterion. For two artificial two-dimensional data sets, the support vectors that are autonomously selected enclose the training set in the input feature space. In the case of more than one cluster in input space, with an ideal mapping, SVDD maps all instances to one spherical-shaped cluster in the transformed feature space, as shown for the two clusters in Fig. 9b.

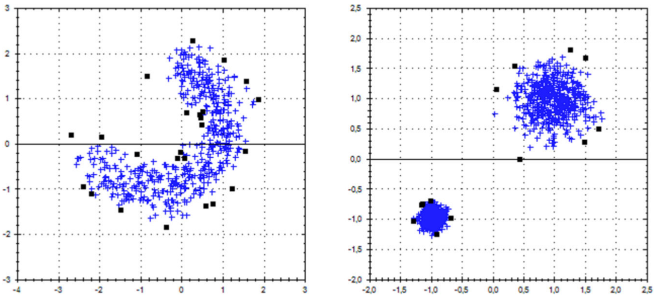


Fig. 9. Parameter tuning of SVDD on artificial two-dimensional data sets visualised in input feature space. (a) banana-shaped cluster and (b) two circular clusters following a Gaussian distribution with different densities (black squares: support vectors).

V. Experimental results

Using grid search, the parameter range was linearly split into 10 candidates, i.e. for the two parameters a grid of 100 (10 × 10) candidate pairs $\{C_i, \sigma_i\}$ was selected per iteration. The parameter ranges were refined by 10 iterations, summing up to 1000 steps. Within each step k-fold is conducted with k = 10. The development of the ranges w.r.t. the optimisation steps are shown in Fig. 10.

The used "banana" data set was created with PRTools [17], the "2 clusters" data set is an artificial data set, where the normal class comprises of two clusters with different densities following a Gaussian distribution and the abnormal class is a ring around the clusters. Since widely used, Fisher's four-dimensional "Iris" data set was utilised. The class "versicolour" was taken as the normal class, where the training was conducted on the first 50% of instances. The "thyroid" data set was taken from [18] and contains 21 features, that are used to determine whether a patient is normal or suffers from either hyperthyroidism or hypothyroidism.

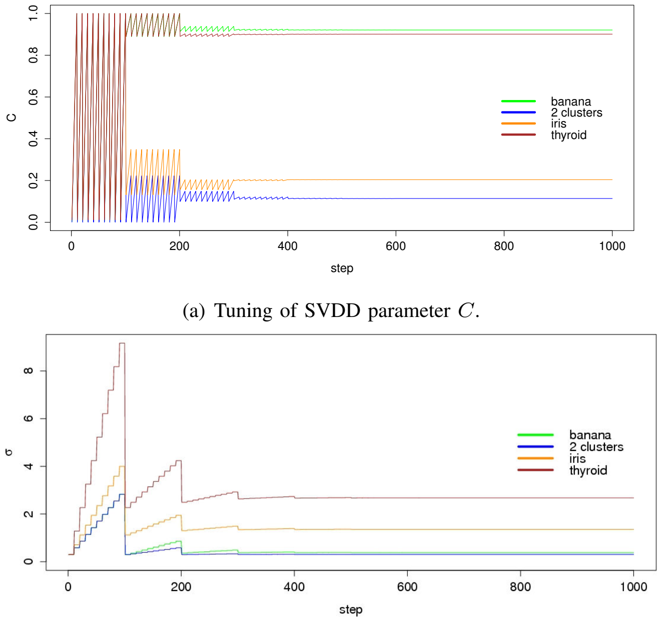
As depicted in Fig. 11, the error rate converges to its minimum, while the radius takes on values close to 1. Consequently the optimisation parameter λ from eq. (31) rapidly converges to its minimum as shown in Fig. 11(c).

The influence of the SVDD parameters C and σ on the solution is summarised as follows:

$$C\uparrow \quad\Rightarrow\quad \begin{cases} R\uparrow \\ e_{\omega_n}, SVs, \lambda\downarrow \end{cases} \tag{32}$$

$$\sigma\uparrow \quad\Rightarrow\quad \begin{cases} \lambda\uparrow \\ e_{\omega_n}, SVs, R\downarrow \end{cases} \tag{33}$$

The proposed parameter tuning approach was experimentally evaluated using selected public domain and own data sets. In addition to the above-mentioned data sets the following data sets were used. The "wine" data set [18] has 13 features determined by chemical analysis of Italian wines from three different cultivars. The first cultivar was used as the normal class. For the "vehicle" data set [18], the task is to classify four types of vehicles based on 18 features extracted from their silhouettes viewed from different angles. The class "van" was used as the normal class. The "DC motor" data set consists of own recordings from a DC motor test rig was used, where anomalies were injected by altering the motor's load.



(a) Tuning of SVDD parameter C.

(b) Tuning of SVDD parameter σ.

Fig. 10. Tuning of SVDD parameters using grid search on the "banana", "2 cluster", "Iris", and "thyroid" data set.

TABLE I
Properties of the used data sets.

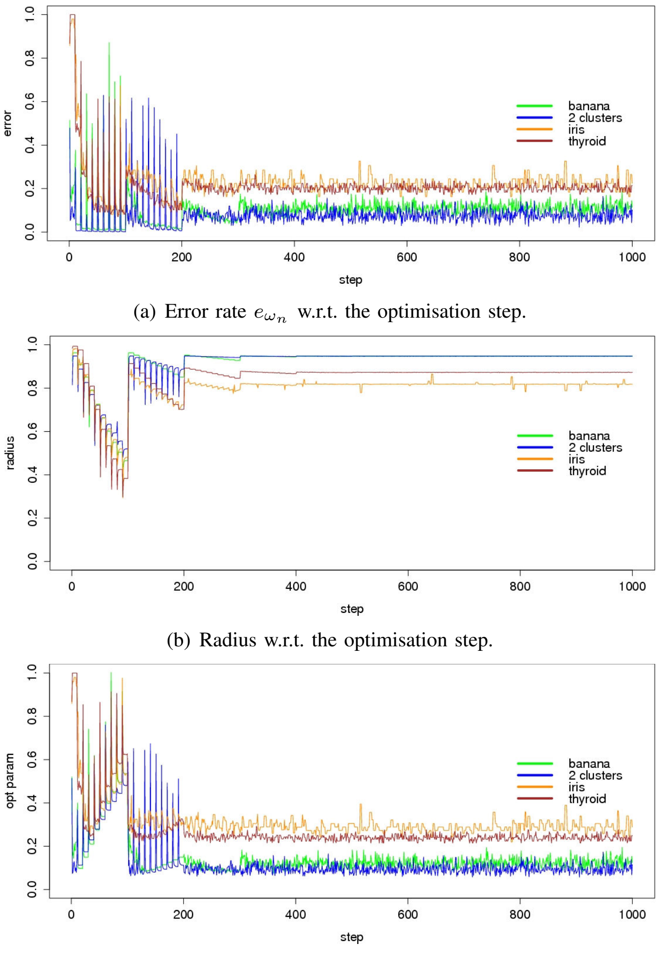
data set	$\|F\|$	$\|A\|$	$\|B\|$: ω_n/ω_a
banana	2	490	489/510
2 clusters	2	1000	1000/2000
Iris	4	25	25/100
wine	13	40	19/119
vehicle	18	122	77/269
thyroid	21	100	66/7034
DC motor	5	4651	18936/482

The properties of all used data sets are summarised in Table I, where $\|F\|$ is the number of features, $\|A\|$ refers to the number of instances in the training set, and $\|B\|$ to the number of instances in the test set respectively.

For all data sets, the features were individually normalised to $+/-1$ prior to classification. The normalisation factors were determined solely from the training set and used to normalise the training and the test set. This way the test set remains a real *blind* test set.

Since the proposed parameter tuning approach was to be validated, no further steps to improve the classification results were taken. In applications, feature selection and feature reduction like PCA should be conducted.

The results are shown in Table II, where SV is the number of determined support vectors, e_{ω_n} the error on the normal class, and R the hypersphere's radius. The last columns hold the true positive rate, the true negative rate and the precision on the abnormal class $\frac{TN}{TN+FN}$.



(a) Error rate e_{ω_n} w.r.t. the optimisation step.

(b) Radius w.r.t. the optimisation step.

(c) Optimisation parameter λ w.r.t. the optimisation step.

Fig. 11. Error rate e_{ω_n}, radius R, and optimisation parameter λ for four data sets evolving over the 1000 tuning steps.

TABLE II
RESULTS ON SELECTED DATA SETS WITH SVDD AND THE PROPOSED AUTONOMOUS PARAMETER TUNING APPROACH.

data set	SVs	e_{ω_n}	R	TPR	TNR	$prec_{\omega_a}$
banana	23	0.037	0.939	95.1%	96.3%	95.3%
2 clusters	36	0.033	0.939	92.7%	99.2%	96.5%
Iris	7	0.150	0.777	92.0%	96.0%	98.0%
wine	11	0.250	0.837	73.7%	100.0%	96.0%
vehicle	20	0.150	0.884	87.0%	82.9%	95.7%
thyroid	17	0.160	0.857	71.2%	96.6%	99.7%
DC motor	20	0.005	0.925	99.5%	76.1%	80.5%

As indicated by the high percentage rates for TNR, the vast majority of anomalies were detected. In addition the precision on the abnormal class is high, i.e. a high fraction of the reported anomalies were indeed abnormal. In addition, as indicated by the low number of support vectors, the knowledge base is stored in a very compact way. This makes the approach applicable for anomaly detection systems.

VI. Conclusion

For anomaly detection problems, where either no abnormal data sets or only a non-representative training data set of anomalies is available, SVDD is an ideal approach. However, for its applicability in real-world applications, the ease of use is crucial. SVDD has the drawback of not being able to automatically tune parameters using cross-validation.

In this paper, a parameter tuning approach for SVDD with the RBF kernel based on an approximation was introduced. It finds the optimal parameter set based on the error rate and the radius, according to the motivation solely on the training set and without the need for the user to adjust parameters. The approach was successfully validated by results on artificial and real data sets.

The proposed approach could make SVDD applicable for a wide range of applications. It is successfully used in the authors' research project on detecting anomalies in recordings from test drives, where users should not be burdened with manual parameter adjustment.

Acknowledgment

The authors are grateful to Oleksandr Pavlichenko, whose findings in his Bachelor thesis inspired this paper, to Gunnar Niess for support with implementation and insightful discussions, and to IT-Designers for funding the research.

References

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, September 2009.

[2] A. Theissler and I. Dear, "Detecting anomalies in recordings from test drives based on a training set of normal instances," in *Proceedings of the IADIS International Conference Intelligent Systems and Agents 2012 and European Conference Data Mining 2012. IADIS Press, Lisbon.*, 2012, pp. 124–132.

[3] A. Theissler and I. Dear, "An anomaly detection approach to detect unexpected faults in recordings from test drives," in *Proceedings of the WASET International Conference on Vehicular Electronics and Safety 2013, Stockholm (to be published).*, 2013.

[4] V. Chandola, "Anomaly detection for symbolic sequences and time series data," Ph.D. dissertation, Computer Science Department, University of Minnesota, 2009.

[5] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, vol. 22, p. 2004, 2004.

[6] S. Theodoridis and K. Koutroumbas, *Pattern Recognition, Fourth Edition*, 4th ed. Academic Press, 2009.

[7] S. Abe, *Support Vector Machines for Pattern Classification (Advances in Pattern Recognition)*, 2nd ed. Springer-Verlag London Ltd., 2010.

[8] D. M. Tax and R. P. Duin, "Data domain description using support vectors," in *Proceedings of the European Symposium on Artificial Neural Networks*, 1999, pp. 251–256.

[9] T. Fawcett, "ROC graphs: Notes and practical considerations for researchers," HP Laboratories, Tech. Rep., 2004.

[10] D. Tax and R. Duin, "Support vector data description," *Machine Learning*, vol. 54, no. 1, pp. 45–66, Jan. 2004.

[11] L. Zhuang and H. Dai, "Parameter optimization of kernel-based one-class classifier on imbalance learning," *Journal of Computers*, vol. 1, no. 7, pp. 32–40, 2006.

[12] D. M. Tax and R. P. Duin, "Uniform object generation for optimizing one-class classifiers," *Journal of Machine Learning Research*, vol. 2, pp. 155–173, 2001.

[13] D. Tax and R. Duin, "Outliers and data descriptions," in *In Proceedings of the Seventh Annual Conference of the Advanced School for Computing and Imaging (ASCI)*, 2001.

[14] D. M. Tax, "One-class classification. concept-learning in the absence of counter-examples," Ph.D. dissertation, Delft University of Technology, 2001.

[15] C. A. Jones, “Lecture notes: Math2640 introduction to optimisation 4,” University of Leeds, School of Mathematics, Tech. Rep., 2005.
[16] O. Pavlichenko, “Adaptation of measured data analysis algorithms for an existing machine learning framework,” 2011.
[17] PRTools, “Website: PRTools: The Matlab Toolbox for Pattern Recognition,” Nov. 2012. [Online]. Available: http://www.prtools.org
[18] KEEL, “Website: KEEL (Knowledge Extraction based on Evolutionary Learning),” Nov. 2012. [Online]. Available: http://sci2s.ugr.es/keel/datasets.php