Expert Systems With Applications 85 (2017) 25–32

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Lightly trained support vector data description for novelty detection

Rekha A.G.[a,*], Mohammed Shahid Abdulla[a], Asharaf S.[b]

[a] IT & Systems Area, Indian Institute of Management Kozhikode, 673570, Calicut, Kerala, India
[b] Indian Institute of Information Technology and Management-Kerala, 695581, Trivandrum, India

ARTICLE INFO

Article history:
Received 3 December 2015
Revised 10 April 2017
Accepted 5 May 2017
Available online 5 May 2017

Keywords:
SVDD
Outlier detection
One-class classification
Scaling

ABSTRACT

Anomaly (or outlier) detection is well researched objective in data mining due to its importance and inherent challenges. An outlier could be the key discovery to be made from large datasets and the insights gathered from them could be of significance in a wide variety of domains like information security, business intelligence, clinical decision support, financial monitoring etc. Recently, Support Vector Data Description (SVDD) driven approaches are shown as having good predictive accuracy. This paper proposes a novel low-complexity anomaly detection algorithm based on Support Vector Data Description (SVDD). The proposed algorithm reduces the complexity by avoiding the calculation of Lagrange multipliers of an objective function, instead locates an approximate pre-image of the SVDD sphere's center, within the input space itself. The crux of the training algorithm is a gradient descent of the primal objective function using Simultaneous Perturbation Stochastic Approximation (SPSA). Experiments using datasets obtained from UCI machine learning repository have demonstrated that the accuracies of the proposed approach are comparable while the training time is much lesser than Classical SVDD.

1. Introduction

Outlier detection, the problem of finding patterns in data that do not conform to expected behavior has attracted lot of attention due to its applicability in a wide variety of domains. One-class classification problem is one of the classical problems in data analysis and has got its original application in outlier detection (Bishop, 1994; Ritter & Gallegos, 1997). The difference of one-class classification from conventional two-class or multi-class classification is that the information on only one of the classes (called the target class) will be available for training. One-class classification has been applied in various scenarios like text classification (Liu, Lee, Yu, & Li, 2002), medical analysis (Gardner, Krieger, Vachtsevanos, & Litt, 2006), machine fault detection (Shin, Eom, & Kim, 2005) etc. Moreover, it has also been applied to various business domains like financial credit scoring (Wang, Wang, & Lai, 2005) and supplier selection (Guo, Yuan, & Tian, 2009). One classical approach for One-Class Classification is Support Vector Data Description (SVDD). SVDD algorithm has been used in scenarios where single class information is available in high quality and resolution, and a few outliers exist. SVDD has also been applied to cases where the problem has to scale to a multi-class environment with information of other classes only gradually becoming available, e.g. in Munoz-Mari, Bruzzone, and Camps-Valls (2007).

SVDD was proposed by Tax and Duin (2004) to solve the original one-class classification problem. The basic idea is to construct a spherically shaped decision boundary that envelops most of the data of interest, with a smaller set of support vectors describing the boundary. This technique is first motivated without using the concept of support vectors.

Given a set of data points, x_i: i=1:N in the d-dimensional real (or input) space R^d, the objective is to minimize an objective function that depends on the radius R of a sphere and its center a.

$$\begin{aligned} &O(R, a, \xi) = R^2 + C\sum_i \xi_i \\ &\text{s.t. } \|x_i - a\|^2 \le R^2 + \xi_i,\ \xi_i \ge 0\ \forall I \end{aligned} \tag{1}$$

Here the parameter C controls the trade-off between the volume and the errors while ξ_i are slack variables which make the classifier 'soft-margin', i.e. allow some possibility of outliers in the training set. Object z is accepted by the description (i.e. z is within the (a, R) sphere) when the Euclidean distance is s.t. : $\|(z-a)^2\| \le R^2$

1.1. Dual & primal SVDD

Normally, for computational convenience and adaptation to the 'Kernel Trick', (1) is solved in the dual space by introducing its Lagrangian function. A description is given in Tax and Duin (2004) as also (4) below. For now, we assume we have the Lagrangian mul-

* Corresponding author.
E-mail addresses: agrekha64@gmail.com (R. A.G.), shahid@iimk.ac.in (M.S. Abdulla), asharaf.s@iiitmk.ac.in (A. S.).

http://dx.doi.org/10.1016/j.eswa.2017.05.007
0957-4174/

tipliers α_i's corresponding to each pattern x_i. The x_i which have an associated $\alpha_i > 0$ are called support vectors (*SVs*). In particular, the *SVs* with $0 < \alpha_i < C$ are called unbounded *SVs* and the *SVs* with $\alpha_i = C$ as the bounded *SVs*. In all calculations within the dual formulation, patterns x_i appear only in the form of inner products with other patterns (x_i. x_j). These inner products can be replaced by a kernel function *K* to obtain more flexible methods. This kernel function *K*is analogous to inner product in a possibly infinite dimensional hyper-space, and represents the 'kernel trick' of Classical SVDD (C-SVDD). The centre of the minimum enclosing ball a_F and the radius R are represented as

$$a_F = \sum_{i=1}^{N_s} \alpha_i \phi(x_i)$$

$$R^2 = 1 - 2\sum_{x_i \in SVs} \alpha_i K(x_i, x_k) + \sum_{x_i \in SVs}\sum_{x_j \in SVs} \alpha_i \alpha_j K(x_i, x_j)$$

of these the latter quantity is calculable due to K being known. The former quantity is not needed explicitly, the decision function for checking a pattern *z* now becomes:

$$1 - 2\sum_i \alpha_i K(z, x_i) + \sum_{i,j} \alpha_j \alpha_j K(x_i, x_j) \le R^2$$

Thus the testing time complexity for C-SVDD is linear in the number of support vectors. However, solving the dual optimization problem (that yields the lagrange multipliers) is also of high-complexity, typically $O(N^3)$. Studies suggest that primal optimization will be superior for large scale optimization (Chapelle, 2007), due to the observation that when the number of training points N is large, the number of support vectors will also likely be large, and this results in updates of nearly N lagrange multiplier parameters during optimization and a complicated decision function during the testing of the algorithm. Hence it is advisable to directly minimize the primal objective function. We give an reference for this below:

While solving the SVDD problem, (Pauwels & Ambekar, 2011) proposes solving an unconstrained optimization problem in the primal:

$$Minimize \qquad O'_p(a, R) = R^2 + C \sum_{i=1}^{N} (d_i{}^2 - R^2)_+ \tag{2}$$

where $d_i = \|x_i - a\|$ and
$(.)_+$ is the ramp function, i.e. if $X \ge 0$ then $(x)_+ = x$, else $(x)_+ = 0$.

While solving the above, no transformation $\phi(.)$ is applied, and hence the generalization power of the kernel trick is not available in this arrangement.

2. Related work

The C-SVDD discussed above, as well as its variants that rely on expanding spatial resolution at the support vector locations (a method known as Conformal Kernel SVDD or CK-SVDD), as seen in Liu, Weng, Kang, Teng, and Huang (2010) have found applications like the P300 Speller Brain-Computer Interface. The current best complexity to solve the C-SVDD training problem is $O(N)$, an improvement from the original $O(N^3)$ as demonstrated in the core vector application of Chu, Tsang, and Kwok (2004). Even in this work, C-SVDD applies for small cases of the original problem and hence the LT-SVDD algorithm presented here applies there too. Also, the work in Chu et al. (2004) relies crucially for termination on a pre-identified fraction of the expected number of outliers: we do not need this in our algorithm. As is explained in Lee and Wright (2012), while the dual problem in 2-class SVMs is convex, the worst case space complexity is one dual variable per example/pattern. In order to assure that our algorithm does obtain an optimal point, the convexity of the primal problem in SVDD being obtained for a minor modification in Wang, Chung, and Shitong (2011) is our reference. The actual progress towards optimum is done using stochastic gradient methods which are considered popular only in linear SVMs (e.g.Lin (2013)). However, here we introduce an algorithm that adapts stochastic gradient to a method that uses the kernel trick.

3. Proposed work

This work proposes a novel low-complexity anomaly detection algorithm based on Support Vector Data Description (SVDD). For N patterns of dimension d, the current best complexity to solve SVDD training problem is $O(N)$ as demonstrated in Chu et al. (2004). The proposed algorithm reduces the complexity of both training and testing to $O(N + d)$ by avoiding the calculation of the Lagrange multipliers α_i, by locating an approximate pre-image of the SVDD sphere's center in the input space during the training phase itself. The proposed algorithm retains the benefit of the kernel trick: i.e. a minimum enclosing space is more descriptive of the data when calculated in a higher-dimensional feature space. The crux of the training algorithm is a gradient descent of the primal objective function using Simultaneous Perturbation Stochastic Approximation (SPSA) adapted to sub-gradients (He, Fu, & Marcus, 2003) and a recast form of the primal problem suggested in Pauwels & Ambekar (2011) that does away with slack variables.

The rest of this paper is organized as follows. Section 4 reviews the Fast-SVDD (F-SVDD) and then Section 5 describes our proposed procedure LT-SVDD. Experimental results on five UCI benchmark datasets and real-world credit datasets from the literature are presented in Section 6, while Section 7 gives concluding remarks.

4. Fast svdd (f-svdd)

The authors of Liu, Liu, and Chen (2010) propose a method called Fast SVDD (F-SVDD) to reduce the computational burden in the testing phase by replacing the kernel expansion in the decision function by a single kernel term. This work relies on calculating the pre-image $\hat{x}$ of a point termed as the 'agent of the SVDD sphere's centre a_F' and denoted by ψ_a. Note that $\hat{x}$ is in the input space whilst a_F and ψ_a are in the feature space. F-SVDD then uses a simple relationship between ψ_a and a_F, i.e. ψ_a is a scalar multiple of a_F, to re-express the centre with a single vector. Hence the decision function of FSVDD contains only one kernel term, and thus the complexity of the FSVDD decision function during testing is a constant, no longer linear in the support vectors.

F-SVDD solves the pre-image problem to find a pattern $\hat{x} \in R^d$ such that $\psi_a = \phi(\hat{x})$ and $\psi_a = \gamma a_F$. In particular, F-SVDD solves as first step the dual of this problem:

$$\text{Minimize } O_p(R, a_F, \xi_i) = R^2 + C\sum_{i=1}^{N} \xi_i \tag{3}$$

$$\text{Subject to} \quad \|\phi(x_i) - a_F\|^2 \le R^2 + \xi_i, \quad \xi_i \ge 0, \forall i \in \{1..N\}$$

where a_F is the center of the minimum enclosing ball, R is its radius and ξ_i are slack variables that allow the enclosing ball to have a soft margin. Here a_F and the kernel-trick based transformation of input pattern x_i, $\phi(x_i)$, are potentially vectors in the infinite dimensional feature space. All N vectors are assumed to belong to one, non-anomalous, class.

Since it is convenient for computational purposes, it is the dual of this problem that is solved:

$$\text{Maximize } O_d(\alpha) = 1 - \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j K(x_i, x_j) \tag{4}$$

subject to $\sum_i^N \alpha_i = 1$
$0 \le \alpha_i \le C \ \forall i \in \{1..N\} \ C \in \left[\frac{1}{N}, 1\right]$

The next step is solving the optimization problem $\min_{\hat{x}} \|a_F - \phi(\hat{x})\|^2$ where $\hat{x} \in R^m$ is in the space of input patterns. The novelty in Liu et al. (2010) is a closed form for the solution $\hat{x}$ given according to the formula:

$$\hat{x} = \frac{\sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j K(x_i, x_j) x_i}{\sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j K(x_i, x_j)} \tag{5}$$

Once $\hat{x}$ is calculated, the classification of a pattern x is now O (1) since the decision function is now given by $D_f(x) = c' - \frac{2}{\gamma} K(x, \hat{x})$ by employing the Gaussian RBF kernel with $K(x, \hat{x}) = \exp(-\frac{\|x-\hat{x}\|^2}{2\sigma^2})$. Here $c' = 1 - R^2 + 1/\gamma^2$ is a constant where the value of γ is determined as:

$$\gamma = \frac{1}{\|a_F\|} = \begin{cases} \frac{1}{\sqrt{\alpha^T K_k}}, & \textit{if all SVs are unbounded SVs} \\ \frac{1}{\sqrt{\alpha^T K_\alpha}} & \textit{otherwise} \end{cases} \tag{6}$$

This expression for $\hat{x}$ in (5) is, however, calculable only after the training problem is solved in the regular fashion, i.e. by obtaining Lagrangian multipliers α_i in the dual formulation of (4). In the absence of perfect pre-images, $\hat{x}$ will be the best approximate pre-image of the feature space's SVDD hyper-sphere centre. This has been shown in (23) of Liu et al. (2010). It is also important to note that not all points in feature space have a pre-image in the input space.

5. Lightly trained svdd (LT_SVDD)

The current work proposes a training phase of SVDD that learns this pre-image $\hat{x}$ in input space rather than calculating the Lagrange multipliers α_i, thereby reducing the computational complexity of the solution to the SVDD optimization problem. The proposed work uses the primal variant of the SVDD problem (1), and also employs penalty functions and slack variables according to the re-interpretation of the SVDD problem (Pauwels & Ambekar, 2011). In particular, what is obtained is an approximate solution to the SVDD problem. This is since the pre-image of the feature space hypersphere's centre a_F, or even the agent of the centre ψ that belongs to the smooth surface S below, may not exist in input space.

For the Gaussian kernel, the nonlinear mapping ϕ has all continuous derivatives, and all the images will lie on a smooth surface S in feature space F. Also, $K(x, x) = 1, \ \forall x \in R^d$. This indicates that all $\phi(x)$ lie on a unit sphere centered at the origin. Fig. 1 shows the geometrical relationship between the SVDD sphere B_S centered at a_F and the unit sphere B_F centered at O_F in the feature space F induced by the Gaussian kernel. H_F is the hyper plane across the intersection of B_S with B_F. Here ψ_a is the agent-of-the-centre and R' is an updated radius value. Under these conditions the following claim holds.

Theorem 1. Assume that $\hat{x}$ is the pre-image of the agent-of-the-centre ψ_a and a quantity $R' > 0$, If any feature space pattern $\phi(x_i)$ is within the (a_F, R) SVDD hypersphere, then it is also within the (ψ_a, R') hypersphere.

Proof. Consider a point A in the feature space such that ||A-a||=R and also that ||A||= 1. Such an A is at the boundary of B_S and also on B_F. All unbounded SVDDs are candidates for such A's.

Now consider the line segments in feature space $\overline{A\psi_a}$, $\overline{A\phi(x_i)}$ and $\overline{\phi(x_i)\psi_a}$ which make up a triangle. All the three line segments are chords of a circle centered at the feature space's origin O_F. Note that $\overline{A\phi(x_i)}$ and $\overline{\phi(x_i)\psi_a}$ are sides of the inscribed triangle within the arc $\widehat{A\phi(x_i)\psi_a}$ formed by the chord $\overline{A\psi_a}$. Here $\widehat{A\phi(x_i)\psi_a}$ is a minor arc.

Inscribed angle of a minor arc being greater than 90°, $\overline{A\psi_a}$ is the longest side of this triangle.

$\therefore \|A\psi_a\| > \|\phi(x_i)\psi_a\|$ (using $\|\cdot\|_2$ norm)
Also, $\|A\psi_a\| = R'$
$\therefore \|\phi(x_i)\psi_a\| < R' \bullet$

Thus, all the points which will be enclosed by the (a_F,R) sphere will be enclosed by the sphere (ψ_a, R'). Also from the geometric properties discussed in Liu et al. (2010), and the inferences drawn from Fig. 1, we can further infer these results. Note that $\tilde{=}$ represents the closest possible approximation within input space, as per (23) of Liu et al. (2010):

Theorem 2. $\mathbf{a_F \tilde{=} \sqrt{1 - R^2}\phi(\hat{x})}$

Proof. $\langle \psi_a, A - a_F \rangle = 0$ since $\psi_a \perp A - a_F$

$$\Rightarrow \langle \psi_a, A \rangle = \langle \psi_a, a_F \rangle \tag{7}$$

$\|A - a_F\|^2 = R^2$
$R^2 + \|\psi_a - a_F\|^2 = \|A - \psi_a\|^2 \because$ *Pythagoras Thm*
$R^2 + < \psi_a - a_F, \psi_a - a_F > = \langle A - \psi_a, A - \psi_a \rangle$
$R^2 + 1 - 2\langle \psi_a, a_F \rangle + \langle a_F, a_F \rangle = 2 - 2\langle A, \psi_a \rangle$
$R^2 + 1 - 2\langle \psi_a, a_F \rangle + \langle a_F, a_F \rangle \ge 2 - 2\langle a_F, \psi_a \rangle$ by(7)
$\langle a_F, a_F \rangle = 1 - R^2$
$\|a_F\| = \sqrt{1 - R^2}$
$a_F = \sqrt{1 - R^2}\psi_a$

$$a_F \tilde{=} \sqrt{1 - R^2}\phi(\hat{x}) \tag{8}$$

Corollary 2. $(R')^2 = 2(1 - \sqrt{1 - R^2})$

Proof.
$(R')^2 = R^2 + \|\psi_a - \sqrt{1 - R^2}\psi_a\|^2 \because$ *Pythagoras Thm*
$(R')^2 = R^2 + \|\psi_a \ (\ 1 - \sqrt{1 - R^2}\ )\|^2$
$(R')^2 = R^2 + 1 - 2\sqrt{1 - R^2}) + 1 - R^2$
$\therefore \mathbf{(R')^2 = 2(1 - \sqrt{1 - R^2})}$

Using the above, it can be seen that $R^2 = 1 - \alpha^T K \alpha$. This finding can be empirically verified in any implementation of the SVDD Dual problem since $1 - \alpha^T K \alpha$ is the value of the objective function when an RBF Kernel is used.

Using the correction in Chang, Lee, and Lin (2013) that proposes R^2 as a variable, with strong duality to ensure the optimality of the SVDD primal problem, the problem can be stated as:

$$\text{Minimize } O_p(R^2, a_F, \xi_i) = R^2 + C\sum_{i=1}^N \xi_i \tag{9}$$

Subject to $\|\phi(x_i) - a_F\|^2 \le R^2 + \xi_i$
$\xi_i \ge 0 \ , \forall i \in \{1..N\}$
$R^2 > 0$

Now we simplify the problem in (9) above. Note the N constraints $\|\phi(x_i) - a_F\|^2 \le R^2 + \xi_i$, with $\rho = \sqrt{1 - R^2}$, are expressed as:

$$\begin{aligned} \langle \phi(x_i) - a_F, \phi(x_i) - a_F \rangle &= 1 + \langle a_F, a_F \rangle - 2\langle \phi(x_i), a_F \rangle \\ &= 1 + \langle \rho\psi_a, \rho\psi_a \rangle - 2\langle \phi(x_i), \rho\psi_a \rangle \\ &= 1 + \rho^2 - 2\rho\langle \phi(x_i), \psi_a \rangle \\ &\le R^2 + \xi_i \end{aligned} \tag{10}$$

Therefore problem in (9) can be cast as:

$$\text{Minimize } O_p(R^2, a_F, \xi_i) = R^2 + C\sum_{i=1}^N \xi_i \tag{11}$$

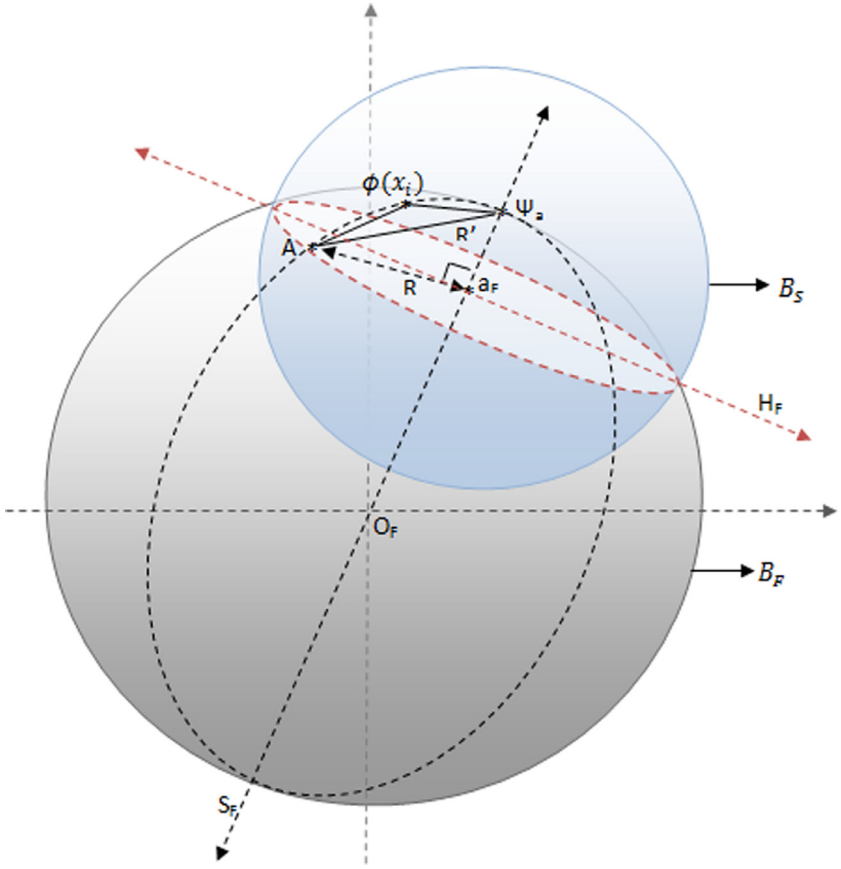

Fig. 1. SVDD sphere B_S and the unit ball B_F in the feature space induced by the Gaussian kernel.

$$\text{Subject to} \quad 1+\rho^2-2\rho\langle\phi(x_i),\,\psi_a\rangle \le R^2+\xi_i$$
$$\psi_a=\left(\tfrac{1}{\rho}\right)a_F$$
$$\xi_i \ge 0\ ,\forall i\in\{1..N\}$$
$$R^2>0$$

Note that (11) is still not soluble due to a_F belonging to possibly infinite-dimensional feature space. A further transformation is possible as below, with optimal R^* and x^* being the required terms R and $\hat{x}$:

$$\text{Minimize } O_p\left(R^2,\,x,\,\xi_i\right)=R^2+C\sum_{i=1}^{N}\xi_i \tag{12}$$

$$\text{Subject to} \quad 1+\rho^2-2\rho K(x_i,x)\le R^2+\xi_i$$
$$\rho=\sqrt{1-R^2}$$
$$\xi_i\ge 0\ ,\forall i\ \in\{1..N\}$$
$$R^2>0$$

5.1. Optimization using SPSA adapted to sub-gradients

As explained earlier, while solving the primal SVDD problem as explained in Pauwels and Ambekar (2011), no transformation $\phi(.)$ was applied, and hence the generalization power of the kernel trick was not available. This is presumably because the authors seek a closed form expression of the *sub-gradient* $\nabla_{a,R}O'_p(a,R)$(since $O'_p(a,R)$ is not everywhere differentiable due to the presence of $(.)_+$ function. They propose a sub-gradient arrangement with $\delta(x)_+$ suitably defined as the *Heaviside step functionH*(.). To summarize, the gradient terms in that work are:

$$\nabla_a O'_p(a,R)=-2C\sum_{i}^{N}H\left(d_i^2-R^2\right)(x_i-a)$$

and

$$\nabla_R O'_p(a,R)=2R-2RC\sum_{i}^{N}H\left(d_i^2-R^2\right)$$

While this is not explicitly mentioned, one assumes that an implementation of their algorithms would involve stochastic approximation of gradient descent as in the two-class method named Approximate Stochastic Subgradient Estimation Training-ASSET (Lee & Wright, 2012). The updates in the method ASSET are as follows:

$$a_{k+1}:=a_k+\beta_k\nabla_a\tilde{O}'_p(a_k,R_k) \tag{13}$$

$$R_{k+1}:=R_k+\beta_k\nabla_R\tilde{O}'_p(a_k,R_k) \tag{14}$$

Where $\nabla_a\tilde{O}'_p(a_k,R_k)$ and $\nabla_R\tilde{O}'_p(a_k,R_k)$ are estimates of the sub-gradient and $\beta_k>0$ is a diminishing step-size with the properties: $\sum_{k=1}^{\infty}\beta_k=\infty$, $\sum_{k=1}^{\infty}{\beta_k}^2<\infty$. This serves to bring iterates a_k and R_k of the algorithm, as $k\to\infty$, towards the optimal $a^*$and R^*. However, no experiments are demonstrated in Pauwels and Ambekar (2011). We also show below how, though the Lagrange multipliers are eliminated, O(N) activity nevertheless takes place due to (13) and (14).

Our proposal relies on the Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm which allows sub-gradient calculation even for complicated optimization problems.

$$\text{Minimize } \hat{O}_p(a,R)=\ R^2+C\ \sum_{i=1}^{N}({d_i}^2-\ R^2)_+ \tag{15}$$

Where $d_i=\|\phi(x_i)-\phi(a)\|$, with $\phi(.)$ as the associated transformation function for the Gaussian Kernel. Note that ${d_i}^2$to be used in $\hat{O}_p(a,R)$ can be calculated as the high-dimensional inner

product $\langle \phi(x_i) - \phi(a), \phi(x_i) - \phi(a) \rangle$ and therefore reduces to (for the particular form of K in Gaussian kernel) $2 - 2.\exp(-\frac{\|x_i - a\|^2}{2\sigma^2})$. A partial gradient (w.r.t the j^{th} component of iterate a viz. a(j)) of the positive portion inside each term of the summation above, i.e. $2 - 2.\exp(-\frac{\|x_i - a\|^2}{2\sigma^2}) - R^2$, can be computed in a closed form. This closed form gradient term would be $\nabla_a^j O_p(a, R) = \frac{-2}{\sigma^2}\exp(\|x_i - a\|^2).(x_i^j - a^j)$, with $a^j$and x^j being j^{th} components of vectors a and x. This would result in a gradient term:

$$\nabla_a \hat{O}_p(a, R) = -\frac{2C}{\sigma^2}\sum_{i=1}^{N} H\left(d_i{}^2 - R^2\right)\exp\left(\|x_i - a\|^2\right)(x_i - a)$$

$$\nabla_R \hat{O}_p(a, R) = 2R - 2RC\sum_{i=1}^{N} H\left(d_i{}^2 - R^2\right)$$

Despite a closed form appearance, the above terms are complicated to calculate due to the value of $h_i := H(d_i^2 - R^2)$ at optimal (a^*, R^*). At each step, d_i is technically $d_i(a_k)$ due to its dependence on the iterate $a_k \in R^m$. Since gradients must equate to 0 at a^*, R^*: values h_i need to have the property that $h_i \in [0, 1]$, $\sum_{i=1}^{N} h_i = \frac{1}{C}$ (from the expression for $\nabla_R \hat{O}_p(a, R)$and further satisfy another condition based on $\nabla_a \hat{O}_p(a, R)$ above). However, this implies that in each iteration k, the h_i values employed need to be computed resulting in O(N) computational complexity. Thus, the Lagrange multipliers α_i may well have been eliminated, but all N patterns need to be evaluated to update a_k, R_k once.

We may instead use an alternative: the work in He et al. (2003) clarifies that SPSA can be used to find the local minima of functions that may not be differentiable everywhere. A perturbation of the iterates a_k and R_k will be used, with the vector $\Delta_a^k (\text{s.t.} \|\Delta_a^k\|_1 = m)$ and scalar Δ_a^k having components drawn with uniform probability from $\{+1, -1\}$. Thus, for a given iterate pair a_k and R_k, we calculate perturbed iterates $a_k = a_k + \delta\Delta_k^a$ and $R_k^+ = R_k + \delta\Delta_k^R$ for a small $\delta > 0$ and obtain an estimate of $O_p^*(a_k^+, R_k^+)$ (call it $\hat{O}_p^*(a_k^+, R_k^+)$). Similarly, we also calculate $a_k^- = a_k - \delta\Delta_k^a$ and $R_k^- = R_k - \delta\Delta_k^R$ and obtain an estimate of the function as $\tilde{O}_p^*(a_k^-, R_k^-)$.

The calculation of the sub-gradient estimate is as follows:

$$\nabla_a \tilde{O}_p'(a_k, R_k)(i) := \frac{\tilde{O}_p^*\left(a_k^+, R_k^+\right) - \tilde{O}_p^*\left(a_k^-, R_k^-\right)}{2\Delta_k^a(i)}$$

$$\nabla_R \tilde{O}_p'(a_k, R_k) := \frac{\tilde{O}_p^*\left(a_k^+, R_k^+\right) - \tilde{O}_p^*\left(a_k^-, R_k^-\right)}{2\Delta_k^R}$$

where $\nabla_a \tilde{O}_p'(a_k, R_k)(i)$ is the i^{th} component of vector $\nabla_a \tilde{O}_p'(a_k, R_k)$. The updates now proceed as in (13) and (14).

The algorithm for LT-SVDD will be as follows:

Algorithm LT SVDD

Training

1. Initialize kernel parameters C and σ using same methods as C- SVDD

2. Solve the pre-image problem for agent of the sphere center by performing optimization in the modified primal (12) to calculate the value of R and $\hat{x}$.

Testing

3. A pattern x_i is treated as typical if $1 + \beta^2 - 2\beta K(x_i, \hat{x}) \leq R^2$

6. Experiments

We have conducted experiments on both synthetic as well as real world data sets to evaluate the suitability of the proposed approach. Since the focus of our study is reducing the complexity of SVDD, we have considered only SVDD based methods for comparison in the initial phase of experiments wherein we have compared the performance using benchmark datasets. C-SVDD and F-SVDD was implemented in MATLAB for comparison. We have done a grid-search for finding adequate values for the parameters C-σ. Further, each combination of the C-σ pair thus obtained is evaluated in terms of the classification performance in C-SVDD. The parameters are selected in accordance to the best performance in terms of C-SVDD accuracy. These values are then used in LT-SVDD which was implemented in C programming language.

Table 1
Datasets used in the experiments.

Dataset	Dimension	Classes	Target class	N Pos	N Neg
Iris	4	3	0	50	100
			1	50	100
			2	50	100
Wine	13	3	0	59	119
			1	71	107
			2	48	130
Cancer	9	2	0	444	239
			1	239	444
Hepatitis	19	2	1	123	32
			2	32	123
Ecoli	7	8	1	193	143
			2	335	1
			3	259	77
			4	334	2
			5	301	35
			6	316	20
			7	331	5
			8	331	5

A fivefold cross-validation method was adopted for the experiments. Each original data set was randomly divided into five stratified parts of equal size. For each fold, four parts have been grouped to form the training data, and the remaining part was used as test set. The comparison of the methods with C-SVDD was done both in terms of the generalization performance to the testing dataset as well as the execution times. We also present a comparison of the performance of our algorithm with the current state of the art ensemble methods including random forests and boosting methods.

6.1. Experiments using standard data sets

For verifying the efficacy of our method, a set of standard datasets from the UCI Machine Learning Repository was used. The datasets considered are IRIS, WINE, CANCER, Hepatitis and Ecoli. These data sets are widely used for comparing classification accuracies (Liu et al., 2010; Dy & Brodley, 2004; Delgado et al., 2014) and are considered as benchmark datasets. In our experiments we take one class as the outlier class, and all other classes will be used as the target class. Further details of these datasets are provided in Table 1.

Our implementations of C-SVDD and primal based SVDD gave comparable accuracies as shown in Table 2. Macro level precision and recall values are calculated using information retrieval concepts as discussed in Dalli (2003) and the results are given in Table 3. The FF-scores calculated using three of the benchmark datasets, i.e. IRIS, WINE AND CANCER are provided in Yang, Sun, and Zhang (2009). Though the work in Yang et al. (2009) is related to clustering, since the final aims are similar we can use their results for comparison and can see that the values of FF-scores are comparable.

6.2. Experiments using credit data sets

Credit scoring is a technique that helps lenders to decide whether or not to grant credit to new applicants based on a number of attributes that describe the socio-demographic and

Table 2
Comparison of testing accuracies.

Dataset	Target class	C_SVDD	F_SVDD	LT_SVDD
Iris	0	98.56 ± 1.36%	97.30 ± 1.80%	99.81 ± 0.42%
	1	93.84 ± 1.05%	91.80 ± 1.66%	92.97 ± 1.98%
	2	93.28 ± 0.92%	93.96 ± 1.57%	92.45 ± 1.01%
Wine	0	93.16 ± 3.25%	93.78 ± 2.79%	95.75 ± 2.67%
	1	83.78 ± 1.52%	82.93 ± 0.84%	85.75 ± 1.26%
	2	96.64 ± 1.53%	95.94 ± 1.36%	97.04 ± 0.58%
Cancer	0	96.20 ± 0.83%	96.00 ± 0.82%	95.84 ± 0.62%
	1	92.68 ± 1.84%	93.26 ± 1.66%	95.19 ± 1.37%
Ecoli	1	81.28 ± 3.72%	77.26 ± 2.05%	83.63 ± 3.31%
	2	97.28 ± 2.29%	93.70 ± 2.17%	97.66 ± 1.61%
	3	75.06 ± 3.33%	64.84 ± 4.59%	74.41 ± 5.45%
	4	92.70 ± 2.89%	88.34 ± 4.25%	96.45 ± 1.10%
	5	66.12 ± 2.89%	64.92 ± 4.25%	64.68 ± 1.10%
	6	82.56 ± 3.09%	73.18 ± 3.62%	77.75 ± 3.35%
	7	91.52 ± 3.32%	88.16 ± 3.42%	91.69 ± 2.18%
	8	93.14 ± 1.80%	87.12 ± 4.53%	92.89 ± 1.88%
Hepatitis	1	89.16 ± 3.11%	88.70 ± 3.69%	88.70 ± 2.31%
	2	70.58 ± 3.54%	65.90 ± 4.43%	70.03 ± 4.64%

Table 3
Precision and recall values.

	Precision	Recall	FF-score
Iris	0.77	0.73	0.75
Wine	0.94	0.69	0.80
Cancer	0.91	0.95	0.93
Ecoli	0.84	0.53	0.65
Hepatitis	0.6	0.64	0.62

Table 4
Test on Japanese dataset.

Model	Accuracy
Linear regression	81.72%
Logit regression	82.53%
Neural network	81.45%
SVM-Linear	80.68%
U-FSVM-Linear	70.27%
B-FSVM-Poly	83.94%
B-FSVM-RBF	83.58%
LT-SVDD	86.56%

Table 5
Test on Australian dataset.

Model	Accuracy
MCLP	75.50%
MCQP	84.50%
RMCLP	89.20%
SVM	88.90%
SVM+ Grid search	85.51%
SVM+Grid search+F-score	84.20%
SVM +GA	86.90%
Bayesian network	85.22%
Naïve Bayes	77.25%
Linear logistic	86.23%
K nearest neighbor	79.42%
C4.5	83.48%
RBF network	83.04%
RIPPER rule induction	85.22%
Ensemble	85.51%
LT-SVDD	84.24%

economic conditions of the applicant. This can be considered as a classification problem in which objective is to classify a credit applicant as creditworthy or not. In credit scoring applications imbalanced class distribution happens and it is a typical example for an outlier detection problem. For our experiments, three real world datasets were obtained from the UCI Machine Learning Repository, namely Japanese, Australian and German credit datasets and was used for evaluating the performance of the proposed method. These datasets have been used previously in many studies to compare the performance of different models (Huang & Wang, 2007; Wang et al., 2005; Shi, Tian, & Zhang, 2009; Leung, Cheong, & Cheong, 2007; Peng, Wang, Kou, & Shi, 2011).

6.2.1. Dataset 1

The first data set used was the Japanese credit dataset. After removing the records with missing attribute values we obtained 653 data, with 357 good cases and 296 bad cases where credit was refused.

6.2.2. Dataset 2

Next, we have tested our algorithm using the Australian credit dataset. It consists of a total of 690 instances with 14 attributes out of which 307 were creditworthy.

6.2.3. Dataset 3

The third data set that we have used was the German dataset with 1000 instances with 700 credit worthy applicants.

7. Results

Tables 4–6 shows the accuracies values obtained from using Japanese, Australian and German data sets respectively, along with accuracy values from the literature. These results indicate that the accuracies of the proposed method is comparable with existing methods.

7.1. Comparison of generalization performance

The accuracy obtained from our implementation of LT_SVDD is compared with the accuracy figures from Linear & Logit Regression, Neural Network, SVM-Linear, U-FSVM-Linear, B-FSVM-Poly, and B-FSVM-RBF taken from Wang et al. (2005) is shown in Table 4.

Table 5 shows the comparison of accuracies for the Australian dataset. In this case, we have taken the accuracies of multiple criteria linear and quadratic program (MCLP & MCQP), regularized multiple criteria linear program (RMCLP) and Support Vector Machine (SVM) from Shi et al. (2009). The accuracies from different variations of SVM was taken from Huang et al. (2007). Further, the accuracies from Bayesian Network, Naïve Bayes, Linear logistic, K nearest neighbor, C4.5, RBF network, RIPPER rule induction, and Ensemble was taken from Peng et al. (2011) for comparison with our method.

Table 6 gives a comparison of accuracy results obtained when German data set was used. Here we have taken the accuracies from MCLP, MCQP, RMCLP and SVM from Shi et al. (2009) and the accuracies from the variations of SVM from Huang et al. (2007). The accuracies of logistic regression (LOG), Linear and quadratic discriminant analysis (LDA & QDA), Neural Networks (NN), Least square SVMs (LS-SVMs), C4.5 decision trees, k-nearest neighbors algorithm (k-NN), Random Forests and Gradient Boosting were taken from Brown and Mues (2012) and the accuracies from Bayesian Network, Naïve Bayes, Linear logistic, K nearest neighbor, C4.5, RBF network, RIPPER rule induction, and Ensemble was taken from Peng et al. (2011).

7.2. Comparison of training times with C-SVDD

Table 7 shows the comparison of classical SVDD with LT-SVDD in terms of execution times and the results shows that LT-SVDD is having superiority over the classical method. The FSVDD requires training time similar to that of C-SVDD.

Table 6
Test on German dataset.

Model	Accuracy
MCLP	66.50%
MCQP	71.50%
RMCLP	72.50%
SVM	73.10%
SVM+ Grid search	76.00%
SVM+Grid search+F-score	77.50%
SVM +GA	77.92%
LOG	76.70%
C4.5	71.20%
NN	72.70%
Gradient boosting	77.20%
LDA	79.10%
QDA	71.8%
Random forests	80.00%
k-NN10	75.0%
k-NN100	79.3%
Lin LS-SVM	81.9%
Bayesian network	72.50%
Naïve Bayes	75.50%
Linear logistic	77.10%
K nearest neighbor	66.90%
C4.5	71.90%
RBF network	74.00%
RIPPER rule induction	73.40%
Ensemble	76.20%
LT-SVDD	70.50%

Table 7
Training times (in sec).

Dataset	N	D	C-SVDD	LT-SVDD
Japanese	653	15	5.54	2.23
Australian	690	14	7.48	1.16
German	1000	24	13.72	3.91

8. Discussion

For this work, we have chosen the credit data sets as an example to examine the applicability of the proposed method to real life data. Class imbalance affects the performance of most the standard classification methods, since they assume a well-balanced class distribution (Japkowicz & Stephen, 2002). Results from previous studies suggest that one-class classifiers perform especially well when data is imbalanced (Kennedy, Namee and Delany, 2009). The data for outlier detection are typically imbalanced since the number of outliers will be very less in number. Literature shows that SVDD based classification methods are well suited for imbalanced data sets like in the case of credit scoring (Tian, Nan, Zheng, & Yang, 2010). Our experimental results indicate that the proposed approach is promising in applications like credit scoring. The results also indicate that it is unlikely to find a single classifier achieving the best results for the whole application domain. Classifiers with best accuracy rate for one data set may have a lower accuracy for another data set. For example, LT-SVDD gives the best accuracy for Japanese dataset (86.56%), while RMCLP gives best accuracy for Australian (89.20%) and Lin LS-SVM for the German (81.9%) data set.

The implications of the proposed method to practitioners are vast. For example, compared to the traditional credit scoring, which is done by banking professionals, an expert system for credit scoring has some obvious advantages like saving cost and time for evaluation of new applications and has important commercial implications. It can also be applied to various other domains requiring detection of outliers. The proposed method helps in automating such requirements with less complexity compared to Classical SVDD approach, especially when faster execution times are desired.

However, this method is also not free from weaknesses. Given the Gaussian kernel, the decision hypersphere of LT-SVDD becomes a sphere in the input space according to the computation of preimage in FSVDD, an illustration of this can be found in (Peng & Xu, 2012). Since LT-SVDD also uses the same idea of *agent of the centre* as in F-SVDD, this drawback is applicable to LT-SVDD too and it may not be immediately applicable to some of anomaly detection cases. Another drawback is the dependency of the accuracy prediction on the attributes chosen. For example, the accuracy of credit scoring in the above example sometimes need not be as capable as that of domain experts. This is because the judgments by the experts based on their experience can sometimes outperform the algorithm outputs.

9. Conclusion

In this work we proposed a method which solves the SVDD problem using the primal form and reduces the complexity by locating an approximate pre-image of the SVDD sphere's center during the training phase itself. The use of SPSA allows us to calculate the gradient for primal gradient-descent even if there is no closed form for the first derivative. Experiments on both benchmark and real-world datasets have demonstrated that the proposed method is promising in reducing the training time and obtaining accuracy.

A possible future direction to carry forward the work is using a similar clustering approach as mentioned in (Peng & Xu, 2012). In Peng & Xu (2012) a Kernel Fuzzy C-Means (KFCM) is used to find cluster centres. But still it could be computationally intensive since calculation of lagrange multipliers is involved. An alternative way is to use KFCM in the input space so that the calculation of lagrange multipliers could be avoided and then proceed in a similar way that is explained in Peng & Xu (2012). Another way to extend this work is to investigate the use of the divide and conquer method as discussed in the Core Vector Machine of Chu et al. (2004) so that faster outlier detection can be performed on even larger datasets with less complexity.

References

Bishop, C. M. (1994). Novelty detection and neural network validation. Paper presented at the vision, image and signal processing, *IEEE proceedings*.

Brown, I., & Mues, C. (2012). An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Systems with Applications, 39*(3), 3446–3453.

Chang, W. C., Lee, C. P., & Lin, C. J. (2013). *A revisit to support vector data description (SVDD)* Technical Report.

Chapelle, O. (2007). Training a support vector machine in the primal. *Neural Computation*, 1155–1178.

Chu, C. S., Tsang, I. W., & Kwok, J. T. (2004). Scaling up support vector data description by using core-sets. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on: Vol. 1* (pp. 425–430). IEEE.

Dalli, A. (2003,). Adaptation of the F-measure to cluster based lexicon quality evaluation. In *Proceedings of the EACL 2003 workshop on evaluation initiatives in natural language processing: Are evaluation methods, metrics and resources reusable?* (pp. 51–56). Association for Computational Linguistics.

Dy, J. G., & Brodley, C. E. (2004). Feature selection for unsupervised learning. *Journal of machine learning research, 5*(Aug), 845–889.

Gardner, A. B., Krieger, A. M., Vachtsevanos, G., & Litt, B. (2006). One-class novelty detection for seizure analysis from intracranial EEG. *The Journal of Machine Learning Research, 7*, 1025–1044.

Guo, X., Yuan, Z., & Tian, B. (2009). Supplier selection based on hierarchical potential support vector machine. *Expert Systems with Applications, 36*(3), 6978–6985.

He, Y., Fu, M. C., & Marcus, S. I. (2003). Fast support vector fast support vector. *IEEE Transactions on Automatic Control, 48*(8).

Huang, C., L, Chen, M. , C., & Wang, C. J. (2007). Credit scoring with a data mining approach based on support vector machines. *Expert Systems with Applications, 33*(4), 847–856.

Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent Data Analysis, 6*(5), 429–449.

Kennedy, K., Mac Namee, B., & Delany, S. J. (2009). Learning without default: A study of one-class classification and the low-default portfolio problem. In *Irish Conference on Artificial Intelligence and Cognitive Science* (pp. 174–187). Berlin Heidelberg: Springer.

Lee, S., & Wright, S. (2012). ASSET: Approximate stochastic subgradient estimation training for support vector machines. *Paper presented at the international conference on pattern recognition applications and methods*.

Leung, K., Cheong, F., & Cheong, C. (2007). Consumer credit scoring using an artificial immune system algorithm. *Paper presented at the IEEE congress on evolutionary computation*.

Lin, C.-J. (2013). *Scalable machine learning in distributed environments*. Paper presented at the Talk at the K. U. Leuven Optimization in Engineering Centre.

Liu, B., Lee, W. S., Yu, P. S., & Li, X. (2002). *Partially supervised classification of text documents*. Paper presented at the ICML.

Liu, Y. H., Weng, J. T., Kang, Z. H., Teng, J. T., & Huang, H. P. (2010). An improved SVM-based real-time P300 speller for brain-computer interface. In *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on* (pp. 1748–1754). IEEE.

Liu, Y. H., Liu, Y. C., & Chen, Y. J. (2010). Fast support vector data descriptions for novelty detection. *IEEE Transactions on Neural Networks, 21*(8), 1296–1313.

Munoz-Mari, J., Bruzzone, L., & Camps-Valls, G. (2007). A support vector domain description approach to supervised classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing, 45*(8).

Pauwels, E. J., & Ambekar, O. (2011). One class classification for anomaly detection: Support vector data description revisited. *Paper presented at the 11th industrial conference on data mining (ICDM)*.

Peng, X., & Xu, D. (2012). Efficient support vector data descriptions for novelty detection. *Neural Computing and Applications, 21*(8), 2023–2032.

Peng, Y., Wang, G., Kou, G., & Shi, Y. (2011). An empirical study of classification algorithm evaluation for financial risk prediction. *Applied Soft Computing, 11*(2), 2906–2915.

Ritter, G., & Gallegos, M. T. (1997). Outliers in statistical pattern recognition and an application to automatic chromosome classification. *Pattern Recognition Letters, 18*(6), 525–539.

Shi, Y., Tian, Y., X, C., & Zhang, P. (2009). Regularized multiple criteria linear programs for classification. *Science in China Series F: Information Sciences, 52*(1), 1–9.

Shin, H. J., Eom, D.-H., & Kim, S.-S. (2005). One-class support vector machines—an application in machine fault detection and classification. *Computers & Industrial Engineering, 48*(2), 395–408.

Tax, D. MJ., & Duin, R. PW. (2004). Support vector data description. *Machine Learning, 54*, 45–66.

Tian, B., Nan, L., Zheng, Q., & Yang, L. (2010). Customer credit scoring method based on the SVDD classification model with imbalanced dataset. In Proceedings of the international conference on e-business technology and strategy. *Ottawa, Canada* (pp. 46–60).

Yang, F., Sun, T., & Zhang, C. (2009). An efficient hybrid data clustering method based on K-harmonic means and particle swarm optimization. *Expert Systems with Applications, 36*(6), 9847–9852.

Wang, X. W., Chung, F.-L., & Shitong (2011). Theoretical analysis for solution of support vector data description. *Neural Networks, 24*(4).

Wang, Y., Wang, S., & Lai, K. K. (2005). A new fuzzy support vector machine to evaluate evaluate credit risk. *Fuzzy Systems, IEEE Transactions, 13*(6), 820–831.