Support Vector Data Description (SVDD)
An experimental study for Anomaly detection

Ezukwoke K.I[1], Zareian S.J[2]

[1, 2] Department of Computer Science
Machine Learning and Data Mining
{ifeanyi.ezukwoke, samaneh.zareian.jahromi}@etu.univ-st-etienne.fr
University Jean Monnet, Saint-Etienne, France

Abstract

Support Vector Data Description (SVDD) is a variant of Support Vector Machines (SVM) used for one class classification. It is particularly designed for outlier detection and hence the focus of our paper. In this paper we introduce the SVDD and its use for outlier detection. We briefly introduce multiple kernel learning and apply it to svdd for outlier detection. We perform experiement on synthetic and real datasets to evaluate the peformance of svdd.

Keywords
One-class classification, support vector classifier, support vector data description (SVDD), outlier detection, novelty detection.

1 INTRODUCTION

Support Vector Data Description (SVDD) is a variant of Support Vector Machines (SVM), usually referred to as the **One class SVM** used to detect novel data or outliers [1]. Its objective is to find an hypersphere of the positive class (usually the target class) among the remaining classes. It particularly finds use in applications such as anomaly detection [2, 3, 4, 5, 6] and novelty detection [7, 8, 9].

The SVDD model is trained on the target class only and assumed to understand the boundary (*hypersphere*) of the target class to label the outliers. We begine by introducing the normal SVDD then proceed to describe SVDD with error before kernellizing it.

2 Support Vector Data Description (SVDD)

Support Vector Data Description (SVDD), originally proposed by [1] is a model which aims at identifying a spherically shaped boundary around a dataset. This dataset is the target dataset for which we expect the model to find outliers for during testing or deployment. As a result SVDD can be regarded as a description of the class of interest [10].

2.1 Primal Formulation of SVDD

Given a set of training data $\{\mathbf{x}_i, \mathbf{y}_i\}$ where $\mathbf{x} \in \mathcal{X}$ is the feature vectors and $\mathbf{y}_i \in \mathcal{Y}$, SVDD aims to minimize the radius of a hypershere $\mathbf{R}$ in a linear space subject to the contraint $\|x_i - c\|^2 \leq R^2$. c is the center of the hypersphere. We can write this as an optimization function

$$\begin{cases} \underset{R \in \mathbb{R}, c \in \mathbb{R}^d}{min} \quad R^2 \\ s.t \quad \|x_i - c\|^2 \leq R^2, i = 1, \ldots, n \end{cases} \tag{1}$$

since $\mathbf{w} = \mathbf{c}$ we can rewrite equation (1) as

$$\begin{cases} \underset{R\in\mathbb{R},c\in\mathbb{R}^d}{min} & R^2 \\ s.t & \|x_i - w\|^2 \leq R^2, i = 1,\dots,n \end{cases} \tag{2}$$

We can also prove that this optimization problem can be rewritten as Quadratic problem as

$$\begin{cases} \underset{\mathbf{w},\rho}{min} & \frac{1}{2}\|\mathbf{w}\|^2 - \rho \\ s.t & \mathbf{w}^T\mathbf{x}_i \geq \rho + \frac{1}{2}\|\mathbf{x}_i\|^2 \end{cases} \tag{3}$$

where $\rho = \frac{1}{2}\left(\|\mathbf{c}^2 - R^2\right)$ and $\mathbf{w} = \mathbf{c}$
Proof:

$$\|x_i - c\|^2 \leq R^2 \tag{4}$$
$$\|x_i\|^2 - 2x_i^2c + \|c\|^2 \leq R^2 \tag{5}$$
$$-2x_i^2c \leq R^2 - \|x_i\|^2 - \|c\|^2 \tag{6}$$
$$x_i^2c \geq \underbrace{\frac{1}{2}(\|c\|^2 - R^2)}_{\rho} + \frac{1}{2}\|x_i\|^2 \tag{7}$$

2.1.1 Dual Formulation for SVDD

We can solve this optimization problem from equation (2) using Lagrangian multipliers.

$$\mathcal{L}(\mathbf{w}, R, \alpha) = R^2 + \sum_{i=1}^{N}\alpha_i(\|x_i - \mathbf{w}\|^2 - R^2) \tag{8}$$

We recall the Karush-Kuhn-Tucker (**KKT**) optimality conditions as

$$\nabla_R\mathcal{L} = 0 \tag{9}$$
$$\nabla_w\mathcal{L} = 0 \tag{10}$$

$$\nabla_R\mathcal{L} = 2R - 2\alpha R = 0 \tag{11}$$
$$\alpha = 1 \tag{12}$$

$$\nabla_w\mathcal{L} = -2\alpha x + 2w\alpha = 0 \tag{13}$$
$$w = \frac{\alpha x}{\alpha} \tag{14}$$

From **Representer theorem**, we know that $w = \alpha x$, therefore

$$w = \frac{\alpha x}{\alpha} = \alpha x \tag{15}$$

If we substitute back the solutions of equations (12) and (15) into (30) we have that

$$\mathcal{L}(\alpha) = \sum_{i=1}^{N}\alpha_i(\|x_i - \alpha_j x_j\|^2) \tag{16}$$
$$= \alpha(x - \alpha_j x_j)^T(x - \alpha_j x_j) \tag{17}$$
$$= \alpha(x^Tx - 2\alpha x^Tx + \alpha_i\alpha_j\alpha x^Tx) \tag{18}$$
$$= \alpha x^Tx - 2\alpha_i\alpha_j x^Tx + \alpha_i\alpha_j\alpha x^Tx \tag{19}$$
$$\mathcal{L}(\alpha) = \alpha x^Tx - \alpha_i\alpha_j\alpha x^Tx \tag{20}$$

The Dual formulation of SVDD can now be written as

$$\begin{cases} \underset{\alpha\in\mathbb{R}^N}{max} & \alpha\mathbf{diag}(\mathbf{G}) - \alpha_i\mathbf{G}\alpha_j \\ s.t & \mathbf{e}^T\alpha = 1 \\ s.t & 0 \leq \alpha_i \quad 1 = 1,\dots,N \end{cases} \tag{21}$$

Where $G = x^Tx$. We maximize the objective function of the dual formulation along the α.

2.2 Primal Formulation of SVDD with errors

The Normal formulation of SVDD from equation (1) is a strict formulation without regards to data points that lie on the decision boundary. We consider the case where the training data cannot be seperated without errors [11]. By allowing a permissible error $\xi \geq 0$, we establish a soft-margin classifier as seen in Support Vector Machine (SVM)-soft margin classifier.

We rewrite equation (2) with errors as

$$\begin{cases} \underset{R\in\mathbb{R},c\in\mathbb{R}^d}{min} & R^2 + C\sum_{i=1}^{N}\xi \\ s.t & \|x_i - w\|^2 \leq R^2 + \xi, i = 1,\dots,n \\ s.t & \xi \geq 0 \quad i = 1,\dots,n \end{cases} \tag{22}$$

Where ξ is the vector of slack variables and $C \geq 0$ is the parameter that controls the tradeoff between the volume of the hypersphere and the permitted errors [10]. Which can also be rewritten as a Quadratic problem

$$\begin{cases} \underset{\mathbf{w},\rho}{min} & \frac{1}{2}\|\mathbf{w}\|^2 - \rho + \frac{C}{2}\sum_{i=1}^{N}\xi \\ s.t & \mathbf{w}^T\mathbf{x}_i \geq \rho + \frac{1}{2}\|\mathbf{x}_i\|^2 - \frac{1}{2}\xi \\ s.t & \xi \geq 0 \quad i = 1,\ldots,n \end{cases} \tag{23}$$

where $\rho = \frac{1}{2}\left(\|\mathbf{c}^2 - R^2\right)$

2.2.1 Dual Formulation for SVDD with errors

We follow the steps for solving the dual formulation without error.

We recall the optimization problem from equation (22) and apply the Lagrangian as follows

$$\mathcal{L}(\mathbf{w}, R, \alpha, \xi) = R^2 + \sum_{i=1}^{N}\alpha_i(\|x_i - \mathbf{w}\|^2 - R^2 - \xi)$$

KKT conditions:

$$\nabla_w \mathcal{L} = 0 \tag{24}$$
$$\nabla_R \mathcal{L} = 0 \tag{25}$$
$$\nabla_\alpha \mathcal{L} = 0 \tag{26}$$
$$\nabla_\xi \mathcal{L} = 0 \tag{27}$$

The solution is similar to that of equation (20). So that our optimization problem of dual SVDD with errors becomes,

$$\begin{cases} \underset{\alpha\in\mathbb{R}^N}{max} & \alpha\mathbf{diag}(\mathbf{G}) - \alpha_i\mathbf{G}\alpha_j \\ s.t & \mathbf{e}^T\alpha = 1 \\ s.t & 0 \leq \alpha_i \leq C \quad 1 = 1,\ldots,N \end{cases} \tag{28}$$

Where $G = x^T x$ and $C \leq 1$. We maximize the objective function of the dual formulation along the α.

Algorithm 1: Linear SVDD using **Stochastic Gradient descent**

Input : κ, α_j
Output : α

begin
  $\alpha_j \leftarrow \alpha^{0}$';
  while *not converged* **do**
    for $i \in$ randshuffle$(\{1,\ldots,N\})$ **do**
      for $k \in \{1,\ldots,K\}$ **do**
        $\alpha_{j+1} = \alpha_j - lr\nabla_\alpha L$;
        where $\nabla_\alpha L = diag(\mathbf{G}) - \alpha\mathbf{G}_k$;
        if $\alpha \geq 0$ **then**
          $\alpha \leftarrow 1$
        end
        if $0 \leq \alpha \leq C$ **then**
          $\alpha \leftarrow 0$
        end
      end
    end
  end
  return α
end

2.3 Kernel Formulation

Given a set of training points $\{x_i \forall i = 1,\ldots N\}$ with a feature vector $\mathbf{x}_i \in \mathbb{R}^N$, we map x_i into a non-linear Hilbert space $\mathcal{H}$ [13].

2.3.1 Primal & Dual Formulation without errors

The primal optimization problem for the kernel SVDD can be extended from equation (1) as

$$\begin{cases} \underset{R\in\mathbb{R},c\in\mathbb{R}^d}{min} & R^2 \\ s.t & \|\phi(x)_i - w\|^2 \leq R^2, i = 1,\ldots,n \end{cases} \tag{29}$$

We introduce the Lagragian multipliers to in the above equations such that

$$\mathcal{L}(\mathbf{w}, R, \alpha) = R^2 + \sum_{i=1}^{N} \alpha_i(\|\phi(x)_i - \mathbf{w}\|^2 - R^2) \tag{30}$$

If we apply the **KKT** conditions we obtain

$$\begin{cases} \nabla_R \mathcal{L}, & \alpha = 1 \\ \nabla_w \mathcal{L}, & \mathbf{w} = \alpha x \end{cases} \tag{31}$$

So that

$$\begin{aligned} \mathcal{L}(\alpha) &= \sum_{i=1}^{N} \alpha_i(\|\phi(x)_i - \alpha_j x_j\|^2) \\ &= \alpha(\phi(x) - \alpha_j \phi(x)_j)^T(\phi(x) - \\ &\qquad \alpha_j \phi(x)_j) \\ &= \alpha(\phi(x)^T\phi(x) - 2\alpha\phi(x)^T\phi(x) + \\ &\qquad \alpha_i\alpha_j\alpha\phi(x)^T\phi(x) \\ &= \alpha\phi(x)^T\phi(x) - 2\alpha_i\alpha_j\phi(x)^T\phi(x) + \\ &\qquad \alpha_i\alpha_j\phi(x)^T\phi(x) \\ \mathcal{L}(\alpha) &= \alpha\phi(x)^T\phi(x) - \alpha_i\alpha_j\alpha\phi(x)^T\phi(x) \end{aligned}$$

If we define a kernel κ as $< \phi(x)^T\phi(x) >$, we can rewrite the equation above as

$$\mathcal{L}(\alpha) = \alpha diag(\kappa(x_i, x_j)) - \alpha_i\alpha_j\kappa(x_i, x_j) \tag{32}$$

The dual optimization problem is now given as

$$\begin{cases} \max\limits_{\alpha\in\mathbb{R}^N} \quad \alpha diag(\kappa) - \alpha_i\kappa\alpha_j \\ s.t \quad \mathbf{e}^T\alpha = 1 \\ s.t \quad 0 \le \alpha_i \quad 1 = 1, \dots, N \end{cases} \tag{33}$$

2.3.2 Primal & Dual Formulation with errors

Similarly, the soft-margin SVDD follows from the error formulation in equation (22) such that we map x_i to a non-linear tranformation $\phi(x_i)$. This formulation is given by

$$\begin{cases} \min\limits_{R\in\mathbb{R}, c\in\mathbb{R}^d} \quad R^2 + C\sum_{i=1}^{N}\xi \\ s.t \quad \|\phi(x_i) - w\|^2 \le R^2 + \xi, i = 1, \dots, n \\ s.t \quad \xi \ge 0 \quad i = 1, \dots, n \end{cases} \tag{34}$$

By introducing Lagrangian multipliers, we have that

$$\mathcal{L}(\mathbf{w}, R, \alpha, \xi) = R^2 + \sum_{i=1}^{N} \alpha_i(\|\phi(x_i) - \mathbf{w}\|^2 - R^2 - \xi)$$

From applying the **KKT** conditions, we know that,

$$\begin{cases} \nabla_R \mathcal{L}, & \alpha = 1 \\ \nabla_w \mathcal{L}, & \mathbf{w} = \alpha x \\ \nabla_\xi \mathcal{L}, & \alpha = 0 \end{cases} \tag{35}$$

Substituting back the optimality result of KKT conditions gives us the Wolfe-Dual the maximization problem

$$\begin{cases} \max\limits_{\alpha\in\mathbb{R}^N} \quad \alpha\mathbf{diag}(\kappa) - \alpha_i\kappa\alpha_j \\ s.t \quad \mathbf{e}^T\alpha = 1 \\ s.t \quad 0 \le \alpha_i \le C \quad 1 = 1, \dots, N \end{cases} \tag{36}$$

Where κ is the kernel matrix $\phi(x)^T\phi(x)$ which satisfies mercers condition [12] and $diag(\kappa)$ is the diagonal of the kernel matrix.

We solve this problem using stochastic

gradient descent algorithm.

Algorithm 2: Kernel SVDD using **Stochastic Gradient descent**

Input : κ, α_j
Output : α
begin
  $\alpha_j \leftarrow \alpha^{0}$';
  while *not converged* **do**
    for $i \in \text{randshuffle}(\{1, \dots, N\})$ **do**
      for $k \in \{1, \dots, K\}$ **do**
        $\alpha_{j+1} = \alpha_j - lr\nabla_\alpha L$;
        where $\nabla_\alpha L = diag(\kappa(x_i, x_j)_k) - \alpha\kappa(x_i, x_j)_k$;
        if $\alpha \geq 0$ **then**
          $\alpha \leftarrow 1$
        end
        if $0 \leq \alpha \leq C$ **then**
          $\alpha \leftarrow 0$
        end
      end
    end
  end
  return α
end

2.4 Kernels

We introduce the commonly used kernels and a brief overview of Multiple kernels used. The radius R is computed from the

- **Linear kernel**

$$\kappa(x_i, x_j) = \mathbf{x}_i\mathbf{x}_j^T \tag{37}$$

- **Polynomial kernel**

$$\kappa(x_i, x_j) = (\mathbf{x}_i\mathbf{x}_j^T + c)^d \tag{38}$$

  where $c \geq 0$ and d is the degree of the polynomial usually greated than 2.

- **RBF(Radial Basis Function) kernel**
  Sometimes referred to as the **Guassian kernel**.

$$\kappa(x_i, x_j) = \exp(-\gamma||\mathbf{x}_i - \mathbf{x}_j||^2) \tag{39}$$

  where $\gamma = \frac{1}{2\sigma^2}$.

- **Sigmoid kernel**

$$\kappa(x_i, x_j) = \tanh(\gamma\mathbf{x}_i\mathbf{x}_j^T + c) \tag{40}$$

  where $c \geq 0$ and $\gamma = \frac{1}{2\sigma^2}$.

- **Cosine kernel**

$$\kappa(x_i, x_j) = \frac{\mathbf{x}_i\mathbf{x}_j^T}{||\mathbf{x}_i||||\mathbf{x}_j||} \tag{41}$$

2.5 Muli-kernels

The reason behind the use of multiple kernels is similar to the notion of multiclassification, where cross-validation is used to select the best performing classifier [?]. By using multiple kernels, we hope to learn a different similarity uncovered using single kernels.

We can prove from **Mercer's Theorem** that a kernel is **Positive Definite (PD)** if $\kappa(x_i, x_j) \geq 0$. Hence by performing arithmetic or any mathematical operation on two or more kernel matrix, we obtain a new kernel capable of exploiting different property or similarities of training data.

Given a kernel κ, we prove that κ is PD if

$$\langle u, \kappa u \rangle \geq 0 \tag{42}$$

Proposition: *A symmetric function κ: $\times \chi \rightarrow \mathbb{R}$ is positive definite $\iff \kappa$*
Proof:
Suppose that κ is a kernel which is the inner product of the mapping functions $\big\langle \phi(x_i)\phi(x_j) \big\rangle$. κ is a kernel if its inner product are positive and the solution of $\kappa u = \lambda u$ gives non-negative eigenvalues.

So that,

$$\langle u, \kappa u \rangle = \sum_{i=1}^{N} u_i(\kappa u)_i \tag{43}$$

$$= \sum_{i=1}^{N} u_i \sum_{j=1}^{N} \big\langle \phi(x_i)\phi(x_j)_{\mathcal{H}} \big\rangle u_j \tag{44}$$

Where $\mathcal{H}$ represent the Hilbert space we project the kernel [13].

$$= \left\langle \sum_{i=1}^{N} u_i\phi(x_i), u_j\phi(x_j) \right\rangle_{\mathcal{H}} \tag{45}$$

$$\langle u, \kappa u \rangle = \left\| \sum_{i=1}^{N} u_i \phi(x_i) \right\|_{\mathcal{H}}^2 \geq 0 \qquad (46)$$

Therefore κ is positive definite.

Using this property of the kernel κ we are able to create serveral other kernels including

- **LinearRBF**
  Here we combine two kernels, precisely **Linear and RBF kernel** using their inner product.
  $$\hat{\mathbf{K}}_{\mathbf{linrbf}} = \kappa(x_i, x_j)^T \kappa(x_i, x_l) \qquad (47)$$
- **RBFPoly**
  Here we combine **RBF and Polynomial kernel** using their inner product.
  $$\hat{\mathbf{K}}_{\mathbf{rbfpoly}} = \kappa(x_i, x_j)^T \kappa(x_i, x_l) \qquad (48)$$
- **EtaKernel**
  The **EtaKernel** is a composite combination of **LinearRBF, RBFPoly and RBFCosine** and it is given by
  $$\hat{\mathbf{K}}_{\mathbf{etarbf}} = \hat{\mathbf{K}}_{\mathbf{linrbf}}^T \hat{\mathbf{K}}_{\mathbf{rbfpoly}} + \hat{\mathbf{K}}_{\mathbf{rbfpoly}}^T \hat{\mathbf{K}}_{\mathbf{rbfcosine}}$$

3 EXPERIMENTAL RESULT

3.1 Dataset

3.2 Performance analysis

4 CONCLUSION

References

[1] David MJ Tax and Robert PW Duin. Support vector data description. Machine learning, 54(1):45–66, 2004.

[2] V. Chandola, A. Banerjee, and V. Kumar, Anomaly detection: A survey. *ACM computing surveys (CSUR)*,vol. 41, no. 3, p. 15, 2009.

[3] Akoglu L., Tong H., and Koutra D. Graph based anomaly detection and description: a survey. *Data mining and knowledge discovery*, vol. 29, no. 3, pp. 626–688, 2015.

[4] Sindagi V. A. and S. Srivastava, Domain adaptation for automatic oled panel defect detection using adaptive support vector data description. *International Journal of Computer Vision*, vol. 122, no. 2, pp. 193–211, 2017.

[5] C. You, D. P. Robinson, and R. Vidal. Provable self representation based outlier detection in a union of subspaces. *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*,pp. 1–10., 2017

[6] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, Adversarially learned one-class classifier for novelty detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3379–3388, 2018.

[7] Abati D., Porrello A., Calderara S., and Cucchiara R. AND: Autoregressive novelty detectors. arXiv preprint arXiv:1807.01653, 2018.

[8] Pimentel M.A., Clifton D.A., Clifton L., Tarassenko L. A review of novelty detection. *Signal Processing*, vol. 99, pp. 215–249, 2014.

[9] Markou M., and Singh S. Novelty detection: a reviewpart 1: statistical approaches. *Signal processing*, vol. 83, no. 12, pp. 2481–2497, 2003.

[10] Zheng S. Smoothly approximated support vector domain description. *Pattern Recognition*, vol 49, pp. 55-64, 2016.

[11] Cortes C., Vapnik V.N. Support Vector Networks. *Machine Learning*, vol 20, no. 3, pp. 273-297, 1995.

[12] Vapnick V. Statistical learning theory. John Wiley. Newyork, 1998.

[13] John Shawt-Taylor, Nello Cristianini. Kernel Methods for Pattern Analysis. Cambridge University Press, ISBN:9780511809682, pg47-83, 2011.