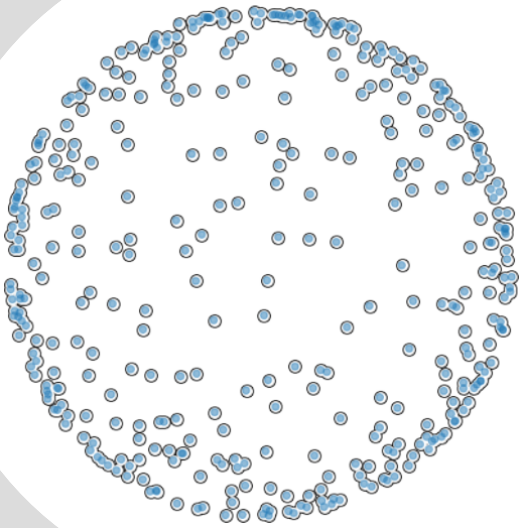
WiSe 2024/25

Machine Learning for Data Science

Lecture by G. Montavon

FREIE UNIVERSITÄT BERLIN



Lecture 4b

Anomalies (cont.)

Outline

Support Vector Data Description (SVDD)

- Primal and Dual formulations
- Recovering/interpreting the Primal from the Dual
- Practical advantage of SVDD over Mahalanobis
- Robustness problem of SVDD

Robustifying SVDD

- Constraint-free reformulation of SVDD
- Robust SVDD: SVDD with linear penalties.
- Practical advantage of Robust-SVDD over SVDD.

Further Applications of Anomaly Detection

Part 1

Support Vector Data Description

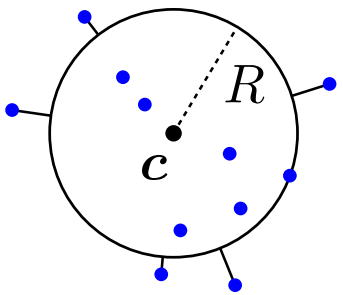
Tax and Duin, Support Vector Data Description.
Machine Learning 54, 45–66, 2004

SVDD: Problem Formulation

- Build a hypersphere

$$\mathcal{H} = \{\boldsymbol{x} \in \mathbb{R}^d \colon \|\boldsymbol{c} - \boldsymbol{x}\|^2 = S\}$$

with parameters $(\boldsymbol{c}, S)$ where S is a variable modeling the squared radius ($S = R^2$). Parameters are chosen in a way that encloses most data points and has minimum radius, i.e. minimizes S.



- Points should be inside the hypersphere. However, to account for the possible presence of anomalies in the data, one allows points to be outside of the hypersphere, but this incurs a penalty ξ_i.
- The problem above can be stated as the convex optimization problem:

$$\min_{S,\boldsymbol{c},\boldsymbol{\xi}} \; S + C\sum_{i=1}^{N} \xi_i \qquad \text{subject to:} \qquad \forall_{i=1}^{N} : \; \|\boldsymbol{x}_i - \boldsymbol{c}\|^2 \leq S + \xi_i$$

$$\forall_{i=1}^{N} : \; \xi_i \geq 0$$

SVDD: From Primal to Dual

Using the framework of KKT conditions, one can derive from the original optimization problem

$$\min_{S,\boldsymbol{c},\boldsymbol{\xi}} S + C\sum_{i=1}^{N}\xi_i \quad \text{s.t.} \quad \forall_{i=1}^{N}: \ \|\boldsymbol{x}_i - \boldsymbol{c}\|^2 \leq S + \xi_i \quad \text{and} \quad \forall_{i=1}^{N}: \ \xi_i \geq 0$$

the *dual* optimization problem:

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{N}\alpha_i\|\boldsymbol{x}_i\|^2 - \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j\boldsymbol{x}_i^\top\boldsymbol{x}_j$$

subject to:

$$\sum_{i=1}^{N}\alpha_i = 1$$

$$\forall_{i=1}^{N}: \ 0 \leq \alpha_i \leq C$$

Observation:

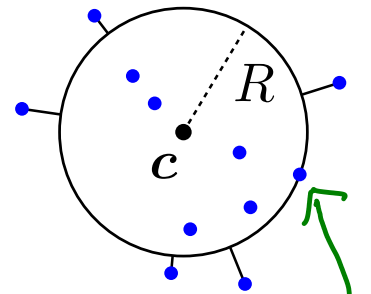
- The original optimization problem has $1 + d + N$ parameters ($\rightarrow$ may not scale well for high-dimensional data). Also the constraints are quadratic ($\rightarrow$ harder to solve than with linear constraints). Instead, the SVDD dual has only N parameters to optimize and has linear constraints.

Recovering Original Parameters from the Dual

- The **center** of the hypersphere can be recovered by the KKT stationary condition $\partial \mathcal{L} / \partial \boldsymbol{c} = \mathbf{0}$, which gives:

$$\boldsymbol{c} = \sum_{i=1}^{N} \alpha_i \boldsymbol{x}_i,$$

  in other words, a weighted average of points in the dataset.



- The **radius** of the hypersphere can be recovered from the KKT complementary slackness equations:

$$\alpha_i \cdot (\|\boldsymbol{x}_i - \boldsymbol{c}\|^2 - S - \xi_i) = 0$$
$$(C - \alpha_i) \cdot (-\xi_i) = 0$$

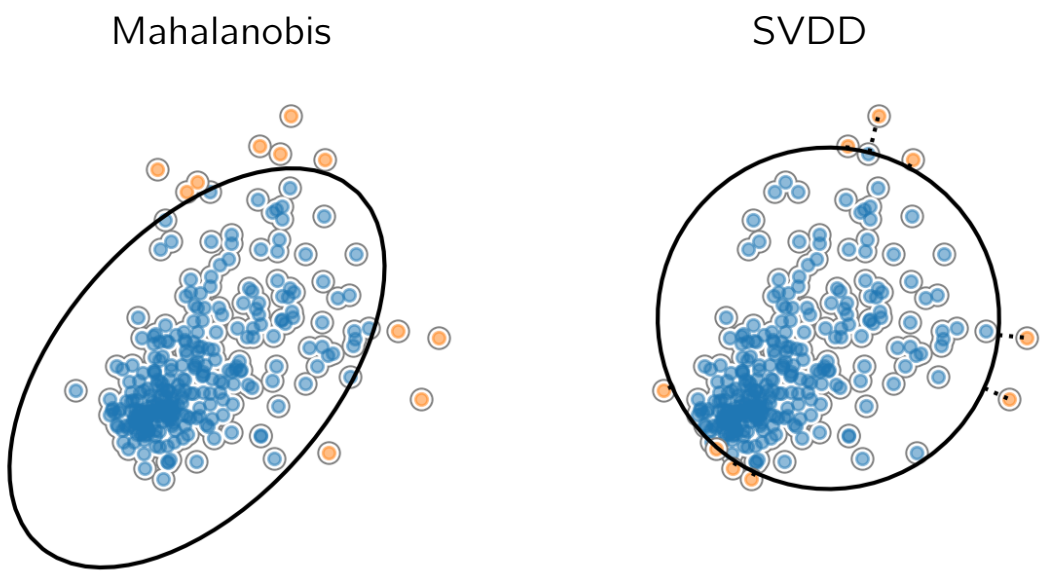
  which yields for any point i satisfying $0 < \alpha_i < C$ the equation, $S = \|\boldsymbol{c} - \boldsymbol{x}_i\|^2$, i.e.

$$R = \|\boldsymbol{c} - \boldsymbol{x}_i\|$$

- We can also infer from these equations that any point inside the hypersphere must have $\alpha_i = 0$, in other words, the hypersphere is solely determined by points at the border of the distribution.

Mahalanobis vs. SVDD in Practice

Example on skewed data:



- The SVDD model is more robust to the skew in the data. This is due to the ability of SVDD to focus mainly on the border of the distribution, and not on the internal distribution within these borders.

Mahalanobis vs. SVDD in Practice

Example on 'high-dimensional' image data:

Mahalanobis (mean vector)

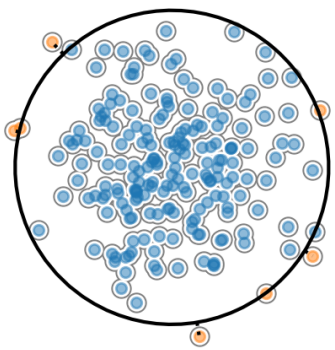
SVDD (center of hypersphere)

Observation:

- SVDD model is more 'blurry' than Mahalanobis. This can be interpreted as SVDD focusing on the border of the distribution, which contains less prototypical, yet non-anomalous, digits.

Is SVDD Robust?

SVDD on normal data



SVDD on poisoned data

Observation:

- The decision boundary of SVDD is greatly distorted by strong outliers. Strong outliers may either be interesting points in the dataset, or some artefacts (e.g. a faults in the data acquisition process or malicious insertions) which we want the model to ignore.
- The algorithm must be robustified to prevent such distortions.

Part 2

Robustifying SVDD

Pauwels et al. One Class Classification for AD: SVDD Revisited
ICDM, 2011

Why is SVDD not Robust?

Diagnosing SVDD:

- Recall that SVDD optimization problem is given by:

$$\min_{S,\boldsymbol{c},\boldsymbol{\xi}} \; S + C\sum_{i=1}^{N}\xi_i \quad \text{s.t.} \quad \forall_{i=1}^{N}: \; \|\boldsymbol{x}_i - \boldsymbol{c}\|^2 \leq S + \xi_i, \quad \forall_{i=1}^{N}: \; \xi_i \geq 0$$

- Penalties associated to outliers (points where $\|\boldsymbol{x}_i - \boldsymbol{c}\|^2 > S$) are given by:

$$\xi_i = \|\boldsymbol{x}_i - \boldsymbol{c}\|^2 - S$$

- The penalty ξ_i grows **quadratically** with the distance of the point $\boldsymbol{x}_i$ from the center $\boldsymbol{c}$. Therefore, the further the outlier from the center, the stronger the center is being pulled towards that outlier.

Question:

- Can we robustify SVDD?

Robustifying SVDD

Step 1:

- Observe that SVDD can also be written as the constraint-free optimization problem:

$$\min_{S,\boldsymbol{c}} \; S + C \sum_{i=1}^{N} \underbrace{\max(0, \|\boldsymbol{x}_i - \boldsymbol{c}\|^2 - S)}_{\xi(\boldsymbol{x}_i)}$$

The added function $\max(0, \cdot)$ handles the two cases where the point is inside/outside the hypersphere. Note that the problem remains convex and its solution can still be found by methods such as gradient descent.

Step 2:

- Consider the alternate convex optimization problem:

$$\min_{R,\boldsymbol{c}} \; R + C \sum_{i=1}^{N} \underbrace{\max(0, \|\boldsymbol{x}_i - \boldsymbol{c}\| - R)}_{\xi^{(\text{new})}(\boldsymbol{x}_i)}$$

Like $\xi(\boldsymbol{x}_i)$, the new penality $\xi^{(\text{new})}(\boldsymbol{x}_i)$ is trigerred exactly at the moment where the data point leaves the hypersphere. However, it grows *linearly* with the distance of the data point from the center. The objective is still convex and can be optimized with gradient descent.

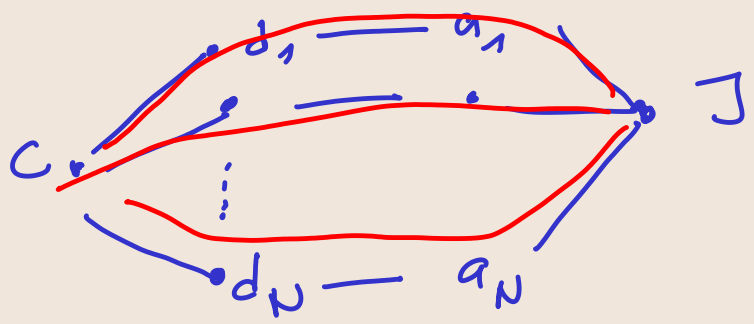
Gradient of Robust SVDD

Exercise: Using the chain rule for derivatives, calculate the gradient of the objective:

$$J(R, \boldsymbol{c}) = R + C \sum_{i=1}^{N} \max(0, \|\boldsymbol{x}_i - \boldsymbol{c}\| - R)$$

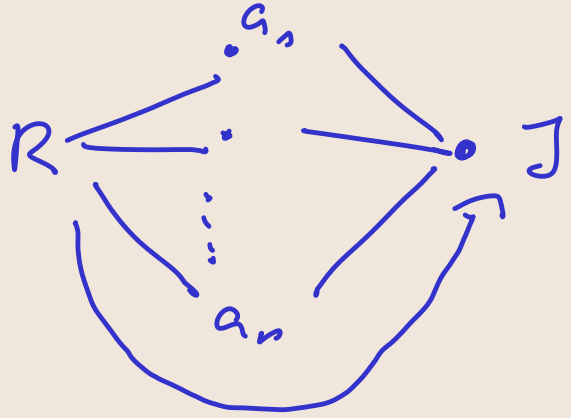
$$\frac{\partial J}{\partial c} = \sum_{i=1}^{N} \frac{\partial J}{\partial a_i} \cdot \frac{\partial a_i}{\partial d_i} \cdot \frac{\partial d_i}{\partial c}$$

$$= \sum_{i=1}^{N} K \cdot 1_{\{d_i > R\}} \cdot \frac{c - x_i}{\|c - x_i\|}$$



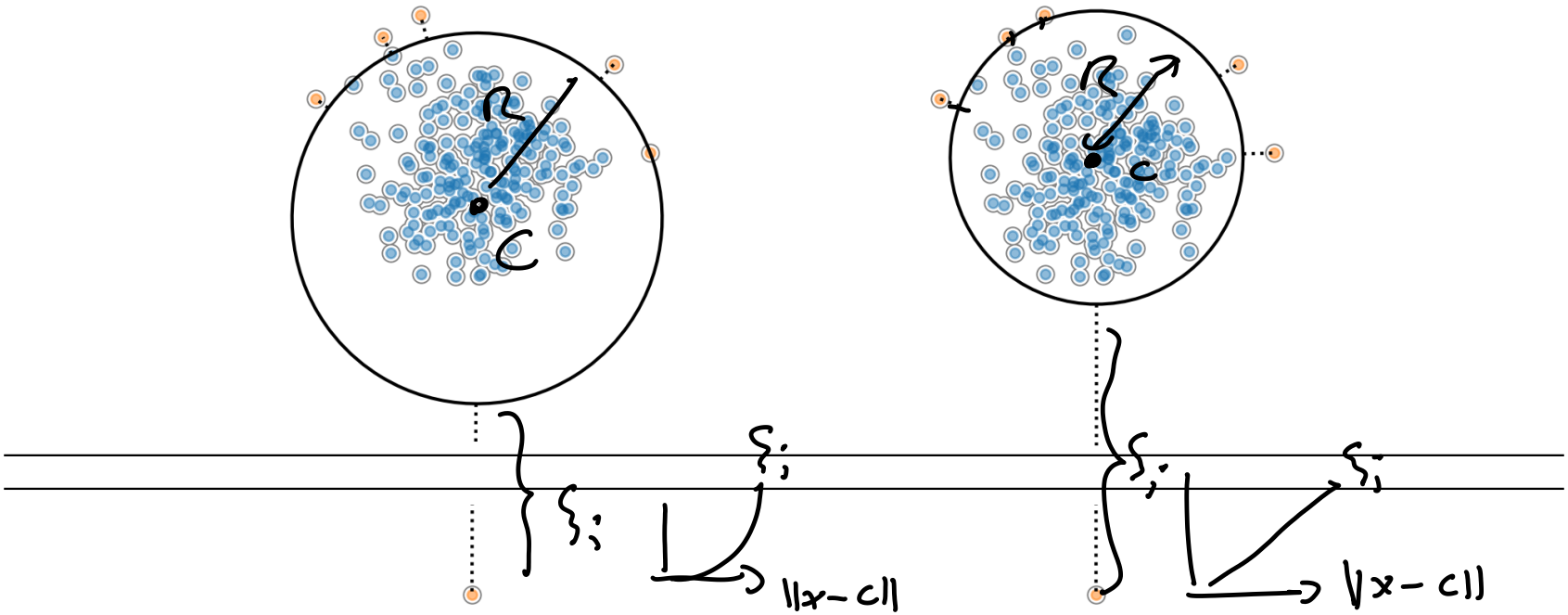
$$\frac{\partial J}{\partial R} = 1 + \sum_{i=1}^{N} \frac{\partial J}{\partial a_i} \cdot \frac{\partial a_i}{\partial R}$$

$$= 1 + \sum_{i=1}^{N} K \cdot 1_{\{d_i > R\}} \cdot (-1)$$



SVDD vs. Robust SVDD

SVDD on poisoned data Robust SVDD on poisoned data



Observation:

- Robust SVDD's decision boundary is left intact, even in presence of a strong outlier.

Part 3

Anomalies Beyond Describing Data

Detecting Flaws in Manufactured Products

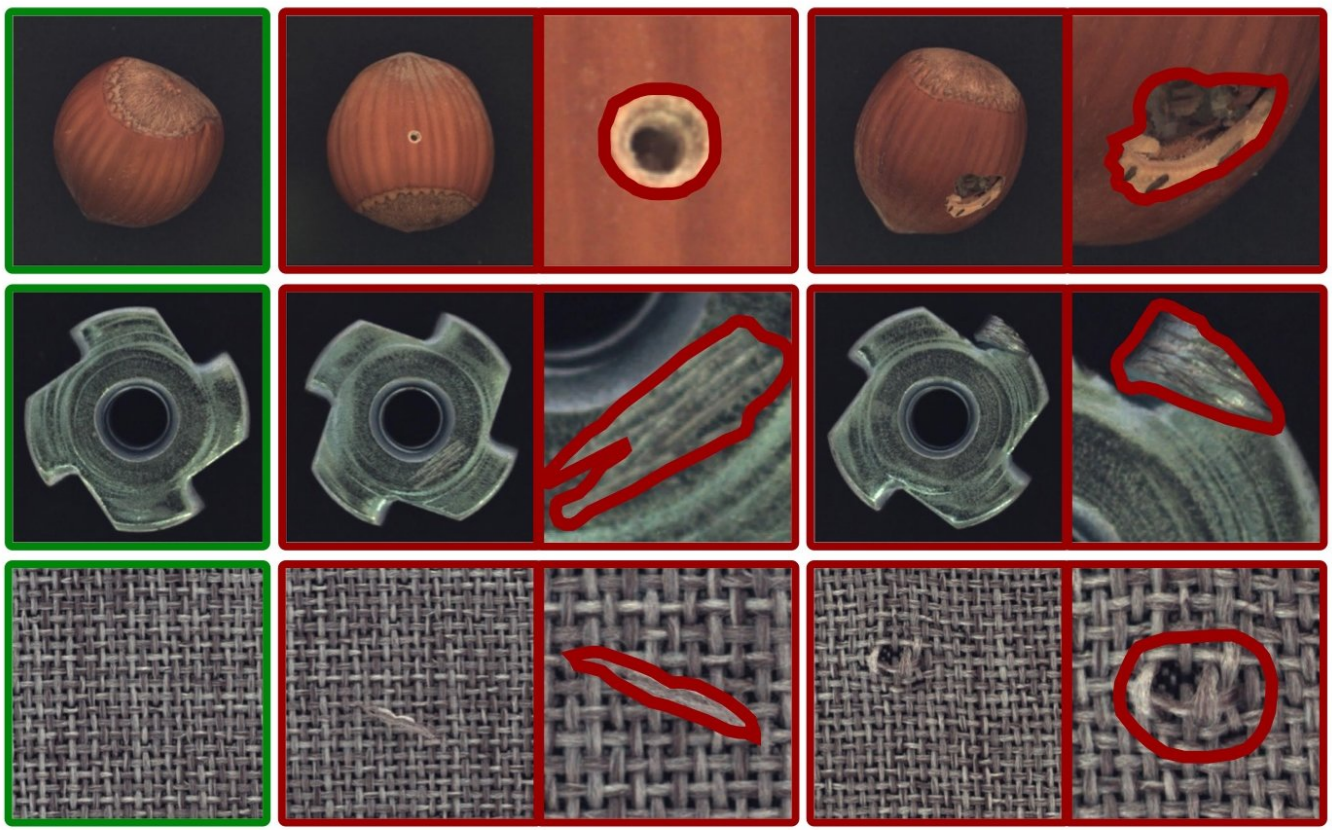
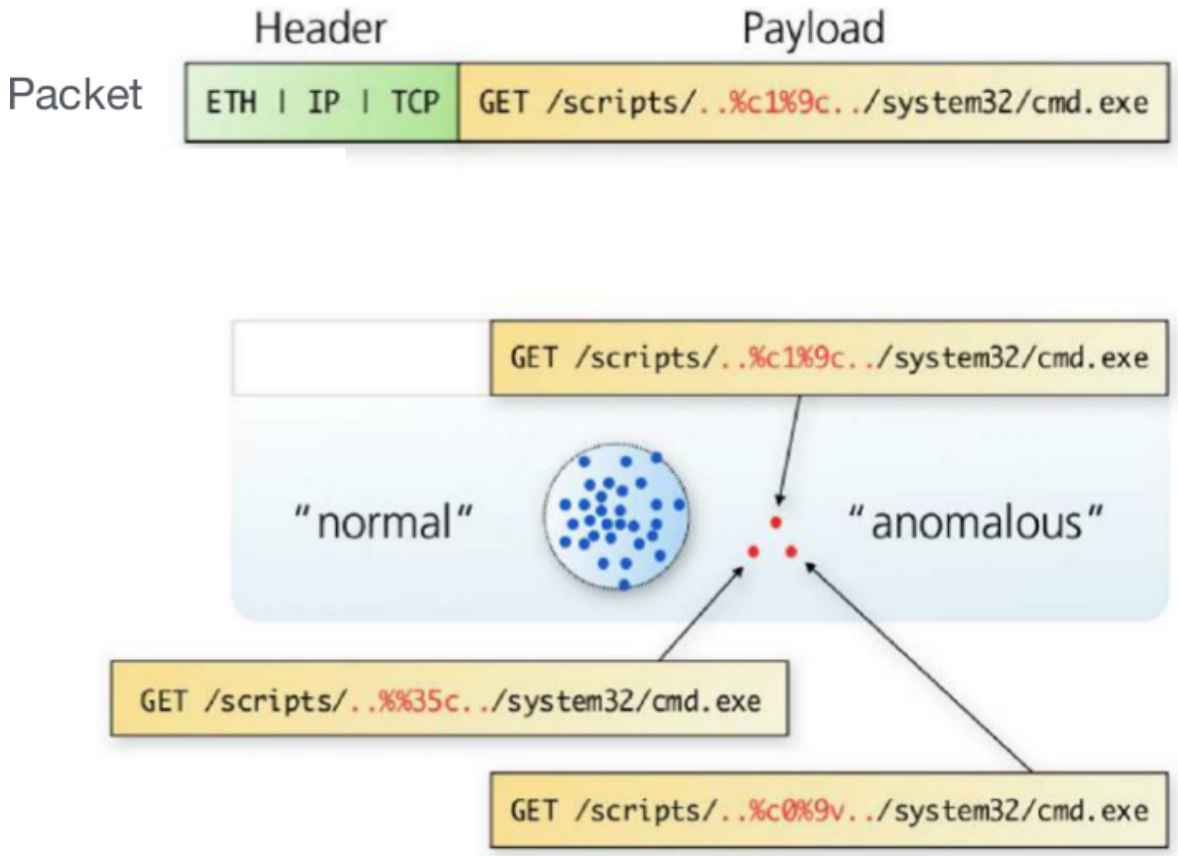


Image source: MvTec

- Anomalous instances can point to flaws in manufacturing.

Detecting Attacks from Network Data



- Anomalous instances can point to execution of malicious code or intrusions.

Summary

Summary

- Support Vector Data Description (**SVDD**) is a popular method for building a boundary between normal and anomalous data. Unlike the Mahalanobis distance, it places the **focus on the border** of the data distribution, and thereby deals better with e.g. skewed data distributions.
- SVDD is formulated as a **convex** optimization problem, and has an associated dual problem. The **SVDD dual** has appealing computational properties. It also allows to better understand the parameters $R, \boldsymbol{c}$ of the original SVDD formulation.
- SVDD lacks robustness to strong outliers. The SVDD method can be **robustified** by first converting it to a constraint-free optimization problem, and then adapt the penalty model $\xi(\boldsymbol{x})$ to achieve insensitivity to strong outliers.