Pattern Recognition 133 (2023) 108999

Contents lists available at ScienceDirect

Pattern Recognition

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

Graph-embedded subspace support vector data description

Fahad Sohrab[a,*], Alexandros Iosifidis[b], Moncef Gabbouj[a], Jenni Raitoharju[c,d]

[a] Faculty of Information Technology and Communication Sciences, Tampere University, Tampere FI-33720, Finland
[b] DIGIT, Department of Electrical and Computer Engineering, Aarhus University, Denmark
[c] Programme for Environmental Information, Finnish Environment Institute, Jyväskylä FI-40500, Finland
[d] Faculty of Information Technology, University of Jyväskylä, Finland

ARTICLE INFO

Article history:
Received 4 September 2021
Revised 8 July 2022
Accepted 20 August 2022
Available online 27 August 2022

Keywords:
One-Class classification
Support vector data description
Subspace learning
Spectral regression

ABSTRACT

In this paper, we propose a novel subspace learning framework for one-class classification. The proposed framework presents the problem in the form of graph embedding. It includes the previously proposed subspace one-class techniques as its special cases and provides further insight on what these techniques actually optimize. The framework allows to incorporate other meaningful optimization goals via the graph preserving criterion and reveals a spectral solution and a spectral regression-based solution as alternatives to the previously used gradient-based technique. We combine the subspace learning framework iteratively with Support Vector Data Description applied in the subspace to formulate Graph-Embedded Subspace Support Vector Data Description. We experimentally analyzed the performance of newly proposed different variants. We demonstrate improved performance against the baselines and the recently proposed subspace learning methods for one-class classification.

1. Introduction

Dimensionality reduction has been an important and active research area in the field of machine learning and data science. The aim is to enhance the performance of a specific application by transforming the data from its original feature space to a lower-dimensional subspace. Dimensionality reduction has been used effectively as a tool in applications ranging from traditional data analysis and classification to many modern applications such as video analytics, recommendation system design, and detecting anomalies in computer and social networks [1].

The three main application domains of dimensionality reduction algorithms are feature matching, model interpretation, and data representation [2]. In feature matching, the aim is to find the similarity between two or more objects via a distance metric such as the Euclidean distance [3]. The model interpretation is enhanced by reducing the number of variables in the subspace by dimensionality reduction methods [4]. In data representation applications, dimensionality reduction methods are used to better represent the data in a lower dimensional space for the task at hand [5].

The approaches used for dimensionality reduction can be either supervised or unsupervised. In supervised learning, the algorithm relies mainly on the structure of data, and the mapping function is inferred from a set of labeled training samples. For example, Fisher's Linear Discriminant Analysis (LDA) is an example of a supervised method that exhibits good discrimination qualities. LDA maximizes the between-class scatter and minimizes the within-class scatter. In unsupervised learning, the algorithm does not leverage the information of pre-existing labels. For example, Principal Component Analysis (PCA) is a well-known unsupervised method for dimensionality reduction. PCA extracts the dominant features of a high-dimensional data and represents it by a small number of orthogonal basis vectors, i.e., the principal components. Numerous extensions and applications of PCA and LDA have been proposed in the literature [6,7], and it has been shown that LDA can outperform PCA when the training data set is large [8]. However, for large-scale datasets, the computation and memory problems, particularly for the eigen-decomposition step of LDA, can be cumbersome. The spectral regression-based technique was proposed in [9] for speeding up the eigen-decomposition step of LDA. The spectral regression-based technique consolidates spectral graph analysis and regression to provide an efficient solution to LDA.

In general, the supervised dimensionality reduction approaches work better than unsupervised algorithms if sufficient data are available [2]. However, in real case scenarios, the labeled data may be scarce, noisy, or expensive to collect. In such situations, semi-supervised learning algorithms are preferred [10]. Semi-supervised

* Corresponding author.
E-mail addresses: fahad.sohrab@tuni.fi (F. Sohrab), ai@ece.au.dk (A. Iosifidis), moncef.gabbouj@tuni.fi (M. Gabbouj), jenni.raitoharju@syke.fi (J. Raitoharju).

https://doi.org/10.1016/j.patcog.2022.108999

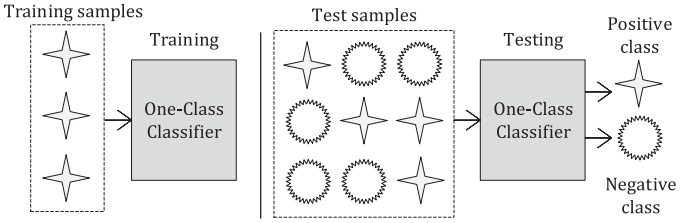

Fig. 1. In one-class classification, a data model is learned by using samples of a positive class only. During inference, the model is used to detect objects also from the negative class.

learning mitigates the necessity for labeled data by allowing a model to leverage unlabeled data. Semi-supervised algorithms can extend the learning strategies of either supervised or unsupervised learning algorithms. If the data are available from only one class during the training, one-class classification algorithms are used to determine the predictive model [11]. In one-class classification, the decision function is inferred using training data from a single class only [12]. The class used to obtain the data description is referred to as the positive class, while all other classes are referred to as the negative class.

One-class classifiers have been extensively studied and improved for several technology-driven applications [13]. One-class classification techniques are found suitable for a specific target class detection in applications such as document classification [14], disease diagnosis [15], fraud detection [16], rare species identification [17], intrusion detection [18], or novelty detection [19]. Fig. 1 depicts the basic idea of one-class classification.

Most one-class classification techniques operate in the original feature space and suffer from the curse of dimensionality [20]. In this paper, we propose a general subspace learning framework for one-class classification. We pose the subspace learning for one-class classification as a graph embedding problem. We show that the previously proposed subspace one-class techniques can be reformulated through the proposed framework, while the framework brings more insight into their optimization process. The framework also allows to integrate other data relations to the optimization process and highlights the similarities to other subspace learning techniques. The framework motivates a novel spectral solution as well as a spectral regression-based solution as alternatives to the previously used gradient-based approach. Finally we integrate the subspace learning framework with the Support Vector Data Description (SVDD) applied in the subspace into an iterative Graph-Embedded Subspace Support Vector Data Description (GESSVDD) method.

The rest of the paper is organized as follows. In Section 2, we review the related work. In Section 3, we formulate the proposed framework, describe the full GESSVDD algorithm, and discuss the new insights obtained from the framework. Details of the experiments and the results are provided in Section 4. We finally deduce the conclusions in Section 5.

2. Related work and background

In this work, we focus on support vector (SV)-based one-class classification methods, which form a decision boundary represented by so-called support vectors by solving an optimization problem. The support vectors are selected from the training data points to define the boundary maximizing the considered criterion uniquely. One-class Support Vector Machine (OCSVM) [21] and SVDD [22] are classic examples of SV-based one-class classification methods. OCSVM constructs a hyperplane that separates the positive class by maximizing the distance of the hyperplane from the origin. In SVDD, a hypersphere with minimum volume is formed around the positive class. Numerous extensions of OCSVM and SVDD have been proposed in the literature [23,24]. Traditionally, the SV-based one-class classification models data in the initially given feature space, but we have recently proposed one-class classification algorithms operating in an optimized lower-dimensional subspace [25,26].

2.1. Support vector data description

SVDD [22] finds a hyperspherical boundary around the positive class data in the original feature space by minimizing the volume of the hypersphere. Let us denote the training samples to be encapsulated inside a closed boundary by a matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N], \mathbf{x}_i \in \mathbb{R}^D$, where N is total number of samples and D is the dimensionality of data. The optimization problem of SVDD is formulated as follows:

$$\begin{aligned} \min \quad & F(R, \mathbf{a}) = R^2 + C \sum_{i=1}^{N} \xi_i \\ \text{s.t.} \quad & \|\mathbf{x}_i - \mathbf{a}\|_2^2 \leq R^2 + \xi_i, \\ & \xi_i \geq 0, \ \ \forall i \in \{1, \ldots, N\}, \end{aligned} \tag{1}$$

where R is the radius and $\mathbf{a} \in \mathbb{R}^D$ is the center of the hypersphere. The slack variables ξ_i, $i = 1, \ldots, N$ are introduced to allow the possibility of data being outliers and the hyperparameter $C > 0$ controls the trade-off between the volume of the hypersphere and the amount of data outside the hypersphere. The Lagrangian of SVDD can be given as

$$L = \sum_{i=1}^{N} \alpha_i \mathbf{x}_i^\intercal \mathbf{x}_i - \sum_{i}^{N} \sum_{j}^{N} \alpha_i \alpha_j \mathbf{x}_i^\intercal \mathbf{x}_j, \tag{2}$$

subject to the constraint that $0 \leq \alpha_i \leq C$ [22]. Maximizing (2) gives a set of α_i values corresponding to each data points. The data points with $0 < \alpha_i < C$ are called *support vectors* and define the data description. A test sample $\mathbf{x}_*$ is classified to the positive class if the distance of the test sample from the center of the hypersphere is smaller than or equal to the radius:

$$\|\mathbf{x}_* - \mathbf{a}\|_2 \leq R, \tag{3}$$

where R is the distance from the center of hypersphere to any sample with $0 < \alpha_i < C$.

2.2. Subspace support vector data description

SSVDD [26] optimizes a data mapping to a lower-dimensional subspace along with data description in the subspace. The optimization function is as follows:

$$\begin{aligned} \min \quad & F(R, \mathbf{a}) = R^2 + C \sum_{i=1}^{N} \xi_i \\ \text{s.t.} \quad & \|\mathbf{Q}\mathbf{x}_i - \mathbf{a}\|_2^2 \leq R^2 + \xi_i, \\ & \xi_i \geq 0, \ \ \forall i \in \{1, \ldots, N\}, \end{aligned} \tag{4}$$

where $\mathbf{Q} \in \mathbb{R}^{d \times D}$ is the projection matrix for mapping the data from original D-dimensional feature space to an optimized lower d-dimensional space. In SSVDD, an iterative process is followed: at each iteration, a set of α_i values is obtained by solving SVDD in the subspace, and then an augmented Lagrangian is optimized to update the projection matrix. The augmented Lagrangian is given as follows:

$$L = \sum_{i=1}^{N} \alpha_i \mathbf{x}_i^\intercal \mathbf{Q}^\intercal \mathbf{Q} \mathbf{x}_i - \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \mathbf{x}_i^\intercal \mathbf{Q}^\intercal \mathbf{Q} \mathbf{x}_j \alpha_j + \beta \psi, \tag{5}$$

where ψ is an optional regularization term expressing the class variance in the lower d-dimensional space and β is the regularization parameter which controls the weight of ψ. The regularization term ψ has the following form:

$$\psi = \mathrm{Tr}(\mathbf{Q}\mathbf{X}\boldsymbol{\lambda}\boldsymbol{\lambda}^\intercal\mathbf{X}^\intercal\mathbf{Q}^\intercal), \tag{6}$$

where Tr is the trace operator and different values of $\boldsymbol{\lambda}$ lead to different variants of SSVDD. The projection matrix $\mathbf{Q}$ is updated by using the gradient of (5), i.e.,

$$\mathbf{Q} \leftarrow \mathbf{Q} - \eta\Delta L, \tag{7}$$

where η is the learning rate parameter. The projection matrix is orthogonalized after every update.

Recently, Ellipsoidal Subspace Support Vector Data Description (ESSVDD) was proposed in [25]. ESSVDD considers the covariance of the data in the subspace and the optimization problem is given as

$$\begin{aligned} \min \quad & R^2 + C\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad & (\mathbf{Q}\mathbf{x}_i - \mathbf{a})^\intercal\mathbf{E}^{-1}(\mathbf{Q}\mathbf{x}_i - \mathbf{a}) \le R^2 + \xi_i, \\ & \xi_i \ge 0, \forall i \in \{1,\dots,N\}, \end{aligned} \tag{8}$$

where

$$\mathbf{E} = \mathbf{Q}\mathbf{X}\mathbf{X}^\intercal\mathbf{Q}^\intercal \tag{9}$$

is a covariance matrix of the data in d-dimensional subspace. The rest of the ESSVDD solution follows the main principles of SSVDD explained above, while including the covariance matrix yields are more generalized solutions compared to SSVDD.

2.3. Graph embedding

Let $\mathcal{G} = \{\mathbf{X}, \mathbf{A}\}$ be an undirected weighted graph, where the data points in $\mathbf{X}$ are the graph nodes and $\mathbf{A} \in \mathbb{R}^{N\times N}$ is the graph weight matrix that can measure different relations between the data points. The Laplacian matrix $\mathbf{L}$ of the graph and the diagonal degree matrix $\mathbf{D}$ are defined as follows:

$$\mathbf{L} = \mathbf{D} - \mathbf{A}, \quad [\mathbf{D}]_{ii} = \sum_{j\neq i}[\mathbf{A}]_{ij}, \forall i \in \{1,\dots,N\}. \tag{10}$$

Graph embedding [27] was proposed as a general framework for encapsulating several subspace learning algorithms under the graph preserving criterion

$$\begin{aligned} \mathbf{Q}^* &= \underset{\mathrm{Tr}(\mathbf{Q}\mathbf{X}\mathbf{L}_p\mathbf{X}^\intercal\mathbf{Q}^\intercal)=m}{\arg\min} \sum_{i\neq j}(\mathbf{Q}\mathbf{x}_i - \mathbf{Q}\mathbf{x}_j)^2\mathbf{A}_{ij} \\ &= \underset{\mathrm{Tr}(\mathbf{Q}\mathbf{X}\mathbf{L}_p\mathbf{X}^\intercal\mathbf{Q}^\intercal)=m}{\arg\min} \mathrm{Tr}(\mathbf{Q}\mathbf{X}\mathbf{L}\mathbf{X}^\intercal\mathbf{Q}^\intercal), \\ &= \arg\min \frac{\mathrm{Tr}(\mathbf{Q}\mathbf{X}\mathbf{L}\mathbf{X}^\intercal\mathbf{Q}^\intercal)}{\mathrm{Tr}(\mathbf{Q}\mathbf{X}\mathbf{L}_p\mathbf{X}^\intercal\mathbf{Q}^\intercal)}, \end{aligned} \tag{11}$$

where $\mathbf{L}$ and $\mathbf{L}_p$ are the graph Laplacian matrices of the *intrinsic* and *penalty* graphs that correspond to data relations to be preserved or penalized, respectively. With different formulations of $\mathbf{L}$ and $\mathbf{L}_p$, (11) can represent different subspace learning algorithms. If there are no data-dependent penalty criteria to consider, the constraint $\mathrm{Tr}(\mathbf{Q}\mathbf{X}\mathbf{L}_\mathbf{p}\mathbf{X}^\intercal\mathbf{Q}^\intercal) = m$ can be replaced with the orthogonality constraint $\mathrm{Tr}(\mathbf{Q}\mathbf{Q}^\intercal) = m$.

The solution to the *trace ratio* optimization in (11) is typically approximated by the corresponding *ratio trace* problem

$$\mathbf{Q}^* = \arg\min \mathrm{Tr}\big((\mathbf{Q}\mathbf{X}\mathbf{L}_p\mathbf{X}^\intercal\mathbf{Q}^\intercal)^{-1}\mathbf{Q}\mathbf{X}\mathbf{L}\mathbf{X}^\intercal\mathbf{Q}^\intercal\big). \tag{12}$$

The solution to (12) can be obtained by solving the generalized eigenvalue value problem

$$\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{q} = \lambda\mathbf{X}\mathbf{L}_p\mathbf{X}^T\mathbf{q} \tag{13}$$

and keeping the eigenvectors corresponding to the d smallest non-zero eigenvalues as the rows of $\mathbf{Q}$.

The total scatter, within-class, and between-classes matrices commonly used in subspace learning can be expressed in the graph embedding framework as follows:

$$\mathbf{S}_t = \mathbf{X}\Big(\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^\intercal\Big)\mathbf{X}^\intercal = \mathbf{X}\mathbf{L}_t\mathbf{X}^\intercal \tag{14}$$

$$\mathbf{S}_w = \mathbf{X}\Big(\mathbf{I} - \sum_{c=1}^{\mathcal{C}}\frac{1}{N_c}\mathbf{1}_c\mathbf{1}_c^\intercal\Big)\mathbf{X}^\intercal = \mathbf{X}\mathbf{L}_w\mathbf{X}^\intercal \tag{15}$$

$$\mathbf{S}_b = \mathbf{X}\Big(\sum_{c=1}^{\mathcal{C}}N_c(\frac{1}{N_c}\mathbf{1}_c - \frac{1}{N}\mathbf{1})(\frac{1}{N_c}\mathbf{1}_c - \frac{1}{N}\mathbf{1})^\intercal\Big)\mathbf{X}^\intercal = \mathbf{X}\mathbf{L}_b\mathbf{X}^\intercal \tag{16}$$

where $\mathbf{I}$ is an identity matrix, $\mathbf{1}$ is a vector of ones, N_c is the total number of instances belonging to class c and $\mathbf{1}_c$ represents a vector with ones corresponding to instances which belongs to class c and zeros elsewhere. For centered data $\mathbf{S}_t$ reduces to $\mathbf{S}_t = \mathbf{X}\mathbf{X}^\intercal$. Using these Laplacians, LDA can be expressed in the graph embedding framework by setting $\mathbf{L} = \mathbf{L}_w$ and $\mathbf{L}_p = \mathbf{L}_b$ in (11). In a similar manner, PCA can be expressed in the graph embedding framework by setting $\mathbf{L} = \frac{1}{N}\mathbf{L}_t$, and replacing the constraint $\mathrm{Tr}(\mathbf{Q}\mathbf{X}\mathbf{L}_\mathbf{p}\mathbf{X}^\intercal\mathbf{Q}^\intercal) = m$ with the orthogonality constraint $\mathrm{Tr}(\mathbf{Q}\mathbf{Q}^\intercal) = m$. Since PCA seeks the projection directions with maximal variances, the criterion is maximized in the case of PCA.

Graph-Embedded Support Vector Data Description [23] was proposed to solve the following optimization problem

$$\begin{aligned} \min \quad & R^2 + C\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad & (\mathbf{x}_i - \mathbf{a})^\intercal\mathbf{S}_x^{-1}(\mathbf{x}_i - \mathbf{a}) \le R^2 + \xi_i, \\ & \xi_i \ge 0, \forall i \in \{1,\dots,N\}, \end{aligned} \tag{17}$$

where $\mathbf{S}_x = \mathbf{X}\mathbf{L}_x\mathbf{X}^\intercal$ and $\mathbf{L}_x$ is the graph Laplacian of any graph expressing geometric data relationship.

2.4. Spectral regression

Spectral regression [28] is an alternative way to solve the generalized eigen-decomposition in (13). If $\mathbf{X}^\intercal\mathbf{q} = \mathbf{t}$, and $\mathbf{t}$ and λ are an eigenvector and eigenvalue solving the eigenproblem

$$\mathbf{L}\mathbf{t} = \lambda\mathbf{L}_p\mathbf{t}, \tag{18}$$

$\mathbf{q}$ is the eigenvector of (13) with the same eigenvalue, because $\mathbf{X}\mathbf{L}\mathbf{X}^\intercal\mathbf{q} = \mathbf{X}\mathbf{L}\mathbf{t} = \lambda\mathbf{X}\mathbf{L}_p\mathbf{t} = \lambda\mathbf{X}\mathbf{L}_p\mathbf{X}^\intercal\mathbf{q}$. In order to find $\mathbf{Q}$, first the target vectors $\mathbf{t}$ can be obtained from (18) and then vectors $\mathbf{q}$ satisfying $\mathbf{X}^\intercal\mathbf{q} = \mathbf{t}$ found. An exact solution may not exists but it can be estimated using regularized least squares also known as ridge regression [29]:

$$\begin{aligned} \mathbf{q} &= \arg\min\Big(\|\mathbf{X}^\intercal\mathbf{q} - \mathbf{t}\|^2 + \eta\|\mathbf{q}\|^2\Big) \\ &= (\mathbf{X}\mathbf{X}^\intercal + \epsilon\mathbf{I})^{-1}\mathbf{X}\mathbf{t}, \end{aligned} \tag{19}$$

where ϵ is a tiny constant. The above technique combines the spectral analysis and the regression, hence the approach is named as spectral regression. The main benefit of spectral regression approach is that most graph Laplacian are sparse and, thus, the approach bypasses the need of computing the eigen-decomposition of dense matrices. The least squares problem can be solved efficiently and, in some cases [9] it is also possible to compute the

target vectors $\mathbf{t}$ directly without using eigen-decomposition at all, which makes the process much faster.

3. Graph embedded subspace support vector data description

In subspace one-class classification, the aim is to determine a projection matrix $\mathbf{Q} \in \mathbb{R}^{d \times D}$ for mapping data $\mathbf{X} \in \mathbb{R}^{D \times N}$ from the D-dimensional original feature space to a lower d-dimensional subspace optimized for one-class classification. In this work, we assume that the data has been centered by setting $\mathbf{X} \leftarrow \mathbf{X} - \boldsymbol{\mu}$, where $\boldsymbol{\mu}$ represents the mean of the training data. The mapped data in the subspace is represented by

$$\mathbf{y}_i = \mathbf{Q}\mathbf{x}_i, \quad i = 1, \ldots, N. \tag{20}$$

After the transformation, the data is encapsulated inside a closed boundary to obtain an optimized data description in the subspace. In order to obtain a generalized solution, we consider the following optimization criterion:

$$\begin{aligned} \min \quad & R^2 + C \sum_{i=1}^{N} \xi_i \\ \text{s.t.} \quad & (\mathbf{Q}\mathbf{x}_i - \mathbf{a})^\intercal \mathbf{S}_Q^{-1} (\mathbf{Q}\mathbf{x}_i - \mathbf{a}) \leq R^2 + \xi_i, \\ & \xi_i \geq 0, \forall i \in \{1, \ldots, N\}, \end{aligned} \tag{21}$$

where the matrix $\mathbf{S}_Q$ encodes geometric data relationships in the subspace as

$$\mathbf{S}_Q = \mathbf{Q}\mathbf{X}\mathbf{L}_x\mathbf{X}^\intercal\mathbf{Q}^\intercal = \mathbf{Q}\mathbf{S}_x\mathbf{Q}^\intercal, \tag{22}$$

where $\mathbf{L}_x$ is a graph Laplacian. It can take different forms depending on the graph type used. By defining a new vector $\mathbf{u} = \mathbf{S}_Q^{-\frac{1}{2}}\mathbf{a}$, (21) can be written as

$$\begin{aligned} \min \quad & R^2 + C \sum_{i=1}^{N} \xi_i \\ \text{s.t.} \quad & \|\mathbf{S}_Q^{-\frac{1}{2}}\mathbf{Q}\mathbf{x}_i - \mathbf{u}\|_2^2 \leq R^2 + \xi_i, \\ & \xi_i \geq 0, \forall i \in \{1, \ldots, N\}. \end{aligned} \tag{23}$$

This shows that we can consider $\mathbf{S}_Q^{-\frac{1}{2}}\mathbf{Q}$ as a new projection matrix to a subspace, where SVDD is to be applied. We denote the mapped input vectors as $\mathbf{z}_i = \mathbf{S}_Q^{-\frac{1}{2}}\mathbf{Q}\mathbf{x}_i$.

The constraints in (23) can be incorporated into a corresponding dual objective function by using Lagrange multipliers:

$$L = R^2 + C \sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \alpha_i \big(R^2 + \xi_i - (\mathbf{S}_Q^{-\frac{1}{2}}\mathbf{Q}\mathbf{x}_i)^\intercal \mathbf{S}_Q^{-\frac{1}{2}}\mathbf{Q}\mathbf{x}_i + 2\mathbf{u}^\intercal \mathbf{S}_Q^{-\frac{1}{2}}\mathbf{Q}\mathbf{x}_i - \mathbf{u}^\intercal\mathbf{u}\big) - \sum_{i=1}^{N} \gamma_i\xi_i, \tag{24}$$

where $\alpha_i \geq 0$ and $\gamma_i \geq 0$ are the Lagrange multipliers. The Lagrangian (24) should be minimized with respect to R, $\mathbf{u}$, and ξ_i and maximized with respect to Lagrange multipliers α_i and γ_i. By setting partial derivative to zero, we get

$$\frac{\partial L}{\partial R} = 0 \Rightarrow \sum_{i=1}^{N} \alpha_i = 1, \tag{25}$$

$$\frac{\partial L}{\partial \mathbf{u}} = 0 \Rightarrow \mathbf{u} = \sum_{i=1}^{N} \alpha_i \mathbf{S}_Q^{-\frac{1}{2}}\mathbf{Q}\mathbf{x}_i, \tag{26}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \Rightarrow C - \alpha_i - \xi_i = 0. \tag{27}$$

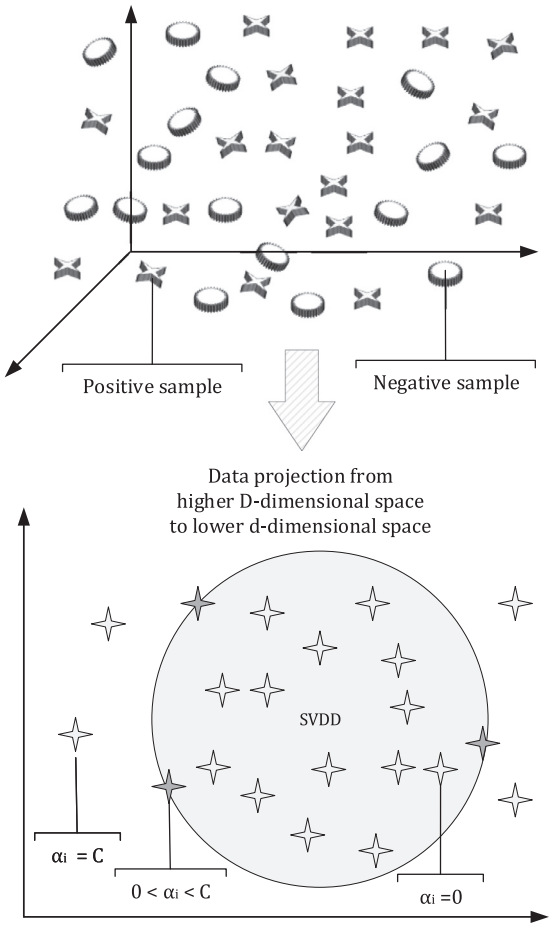

Fig. 2. Depiction of data projection to a lower d-dimensional space optimized for one-class classification with corresponding α_i values.

By substituting (25)-(27) into (24), we get

$$\begin{aligned} L &= \sum_{i=1}^{N} \alpha_i \mathbf{x}_i^\intercal \mathbf{Q}^\intercal \mathbf{S}_Q^{-1} \mathbf{Q}\mathbf{x}_i - \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \mathbf{x}_i^\intercal \mathbf{Q}^\intercal \mathbf{S}_Q^{-1} \mathbf{Q}\mathbf{x}_j \alpha_j \\ &= \sum_{i=1}^{N} \alpha_i \mathbf{z}_i^\intercal \mathbf{z}_i - \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j \mathbf{z}_i^\intercal \mathbf{z}_j. \end{aligned} \tag{28}$$

Maximizing (28) corresponds to solving SVDD in the new subspace and will give us α_i values for all instances, which will define their position in the data description. The samples in the subspace corresponding to values $0 < \alpha_i < C$ will lie on the boundary, while those outside the boundary will correspond to values $\alpha_i = C$. For the samples inside the closed boundary, the corresponding values of α_i will be equal to zero:

$$\|\mathbf{z}_i - \mathbf{u}\|_2 < R \rightarrow \alpha_i = 0, \gamma_i = 0, \tag{29}$$

$$\|\mathbf{z}_i - \mathbf{u}\|_2 = R \rightarrow 0 < \alpha_i < C, \gamma_i = 0, \tag{30}$$

$$\|\mathbf{z}_i - \mathbf{u}\|_2 > R \rightarrow \alpha_i = C, \gamma_i > 0. \tag{31}$$

Fig. 2 depicts the idea of projecting data into an optimized subspace along with the positions of instances according to α values. The negative class samples are not considered in the process; hence, it is not guaranteed that they will be outside the obtained closed boundary.

The Lagrangian in (28) can be written in a trace form as

$$\begin{aligned} L &= \mathrm{Tr}(\mathbf{S}_Q^{-1}\mathbf{Q}\mathbf{X}\mathbb{A}\mathbf{X}^\intercal\mathbf{Q}^\intercal) - \mathrm{Tr}(\mathbf{S}_Q^{-1}\mathbf{Q}\mathbf{X}\boldsymbol{\alpha}\boldsymbol{\alpha}^\intercal\mathbf{X}^\intercal\mathbf{Q}^\intercal) \\ &= \mathrm{Tr}((\mathbf{Q}\mathbf{X}\mathbf{L}_x\mathbf{X}^\intercal\mathbf{Q}^\intercal)^{-1}\mathbf{Q}\mathbf{X}(\mathbb{A}-\boldsymbol{\alpha}\boldsymbol{\alpha}^\intercal)\mathbf{X}^\intercal\mathbf{Q}^\intercal), \end{aligned} \tag{32}$$

where the matrix $\mathbb{A} \in \mathbb{R}^{N\times N}$ contains α_i values in its diagonal and zeros elsewhere, $\boldsymbol{\alpha}$ is a vector of α_i values. Now by defining the matrices

$$\mathbf{L}_\alpha = \mathbb{A} - \boldsymbol{\alpha}\boldsymbol{\alpha}^\intercal \tag{33}$$

$$\mathbf{S}_\alpha = \mathbf{X}\mathbf{L}_\alpha\mathbf{X}^\intercal, \tag{34}$$

we can simplify (32) to

$$L = \mathrm{Tr}\big((\mathbf{Q}\mathbf{S}_x\mathbf{Q}^\intercal)^{-1}\mathbf{Q}\mathbf{S}_\alpha\mathbf{Q}^\intercal\big). \tag{35}$$

We note that (35) is in a ratio trace form that resembles the trace ratio in (11). As mentioned, the trace ratio in (11) is typically approximated by the corresponding ratio trace to be able to solve the optimization using eigen-decomposition. We also note that $\mathbf{L}_\alpha$ is a graph Laplacian (see Section 3.2). Thus, we have presented the subspace learning for SVDD in the general graph embedding framework for subspace learning with its own fixed intrinsic graph $\mathbf{L}_\alpha$. Different graphs $\mathbf{L}_x$ create different variants and can be selected to enforce different constraints for the data. We will get back to different insights offered by the new framework in Section 3.2, but first we will introduce the full Graph-Embedded Subspace Support Vector Data Description (GESSVDD) algorithm.

3.1. GESSVDD Algorithm

We can directly see from (35) that it can be minimized/maximized by solving the generalized eigenproblem in (13) and keeping the eigenvectors corresponding to the smallest/largest non-zero eigenvalues as projection vectors. We can also formulate a spectral regression-based solution as explained in Section 2.4. While earlier subspace SVDD variants [25,26,30] have only used gradient-based solution, we now have three alternatives: 1) gradient-based, 2) spectral, and 3) spectral regression-based updates. Furthermore, we can pick any desired graph as $\mathbf{L}_x$ and we note that it can be meaningful to also maximize (27) (see further discussion in Section 3.2). With this we can give the main GESSVDD algorithm in Algorithm 1 and the three update options in Sub-algorithms 1–3. The gradient of (32) used in the gradient-based update can be obtained using identity 126 in [31].

3.1.1. Non-linear data description

To obtain a non-linear mapping with the proposed method, we employ a non-linear projection trick (NPT) [32]. NPT is equivalent to applying the well-known kernel trick, while allows using the linear variant of the method. In NPT, the data $\mathbf{X}$ is mapped from the original D-dimensional space to $\boldsymbol{\Phi}$ in F-dimensional space as follows: The kernel matrix is obtained as

$$\mathbf{K}_{ij} = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}\right), \tag{36}$$

where σ is a hyperparameter scaling the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. The kernel matrix is centered as

$$\hat{\mathbf{K}} = \big(\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^\intercal\big)\mathbf{K}\big(\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^\intercal\big), \tag{37}$$

The centered kernel matrix $\hat{\mathbf{K}}$ is decomposed by using eigen-decomposition:

$$\hat{\mathbf{K}} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\intercal, \tag{38}$$

Algorithm 1: GESSVDD optimization .

Input : $\mathbf{X}$, // Input data
$\mathbf{L}_x$ // Selected Laplacian
η, // Learning rate parameter
d, // Dimensionality of subspace
C, // Regularization parameter in SVDD
min or max // Either minimize or maximize the criterion

Output: $\mathbf{Q}$ // Projection matrix
R, // Radius of hypersphere
$\boldsymbol{\alpha}$ // Defines the data description

Initialize $\mathbf{Q}$ via PCA; // Select d-vectors corresponding to d largest eigenvalues.
Compute $\mathbf{S}_x = \mathbf{X}\mathbf{L}_x\mathbf{X}^\intercal$;

for $iter = 1 : max\_iter$ **do**
  Calculate $\mathbf{S}_{inv} = \mathbf{S}_Q^{-1} = (\mathbf{Q}\mathbf{S}_x\mathbf{Q}^\intercal)^{-1}$;
  Project data to subspace $\mathbf{z}_i = \mathbf{S}_Q^{-\frac{1}{2}}\mathbf{Q}\mathbf{x}_i = (\mathbf{S}_{inv})^{\frac{1}{2}}\mathbf{Q}\mathbf{x}_i$;
  Calculate α values by maximizing$L = \sum_{i=1}^N \alpha_i\mathbf{z}_i^\intercal\mathbf{z}_i - \sum_{i=1}^N\sum_{j=1}^N \alpha_i\alpha_j\mathbf{z}_i^\intercal\mathbf{z}_j$;
  Compute $\mathbf{L}_\alpha = \mathbb{A} - \boldsymbol{\alpha}\boldsymbol{\alpha}^\intercal$;
  if *gradient-based update*
  Call Sub-algorithm 1 to obtain $\mathbf{Q}$;
  elseif *spectral update*
  Call Sub-algorithm 2 to obtain $\mathbf{Q}$;
  elseif *spectral regression-based update*:
  Call Sub-algorithm 3 to obtain $\mathbf{Q}$;
  endif
  Orthogonalize $\mathbf{Q}$ using QR decomposition;
end
Project data to subspace $\mathbf{z}_i = \mathbf{S}_Q^{-\frac{1}{2}}\mathbf{Q}\mathbf{x}_i$;
Calculate α values by maximizing$L = \sum_{i=1}^N \alpha_i\mathbf{z}_i^\intercal\mathbf{z}_i - \sum_{i=1}^N\sum_{j=1}^N \alpha_i\alpha_j\mathbf{z}_i^\intercal\mathbf{z}_j$;
Compute center of data description in the subspace as $\mathbf{u} = \sum_{i=1}^N \alpha_i\mathbf{S}_Q^{-\frac{1}{2}}\mathbf{Q}\mathbf{x}_i$;
Identify any support vector $\mathbf{s}$ having $0 < \alpha_s < C$;
Compute radius$R = \sqrt{(\mathbf{S}_Q^{-\frac{1}{2}}\mathbf{Q}\mathbf{s})^\intercal\mathbf{S}_Q^{-\frac{1}{2}}\mathbf{Q}\mathbf{s} - 2(\mathbf{S}_Q^{-\frac{1}{2}}\mathbf{Q}\mathbf{s})^\intercal\mathbf{u} + \mathbf{u}^\intercal\mathbf{u}}$;

where $\boldsymbol{\Lambda}$ contains the non-negative eigenvalues of $\hat{\mathbf{K}}$ in its diagonal and the columns of $\mathbf{U}$ contain the corresponding eigenvectors. Finally, the data representation $\boldsymbol{\Phi}$ is obtained as

$$\boldsymbol{\Phi} = \boldsymbol{\Lambda}^{\frac{1}{2}}\mathbf{U}^\intercal. \tag{39}$$

Now we consider the obtained data transformation $\boldsymbol{\Phi}$ as the input to the linear algorithm, which is equivalent to applying the kernel method on $\mathbf{X}$.

3.1.2. Test phase

During testing, a test instance $\mathbf{x}_*$ is first mapped to an optimized d-dimensional space as

$$\mathbf{z}_* = \mathbf{S}_Q^{-\frac{1}{2}}\mathbf{Q}\mathbf{x}_*. \tag{40}$$

The distance of the test instance to the center of the data description in the subspace is calculated. The test instance is classified as a positive instance if the distance is equal to or smaller than the radius:

$$\|\mathbf{z}_* - \mathbf{u}\|_2^2 \leq R^2, \tag{41}$$

where $\mathbf{u}$ is obtained by solving (26), and R^2 is calculated as

$$R^2 = (\mathbf{S}_Q^{-\frac{1}{2}}\mathbf{Q}\mathbf{s})^\intercal \mathbf{S}_Q^{-\frac{1}{2}}\mathbf{Q}\mathbf{s} - 2(\mathbf{S}_Q^{-\frac{1}{2}}\mathbf{Q}\mathbf{s})^\intercal \mathbf{u} + \mathbf{u}^\intercal\mathbf{u}, \tag{42}$$

and $\mathbf{s}$ is any support vector with $0 < \alpha_s < C$. Otherwise, the test instance is classified as a negative instance.

In the non-linear approach, we first find the kernel vector

$$\mathbf{k}_* = \mathbf{\Phi}^\intercal \phi(\mathbf{x}_*). \tag{43}$$

The kernel vector is centered as

$$\hat{\mathbf{k}}_* = (\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^\intercal)[\mathbf{k}_* - \frac{1}{N}\mathbf{K}\mathbf{1}]. \tag{44}$$

Finally, the NPT representation of the test instance is obtained as

$$\boldsymbol{\phi}_* = (\mathbf{\Phi}^T)^+ \hat{\mathbf{k}}_*, \tag{45}$$

where $(.)^+$ is a pseudo-inverse. Now $\boldsymbol{\phi}_*$ is classified similar to the linear case, which is equivalent to applying a kernel method on $\mathbf{x}_*$.

3.1.3. Different variants

While any suitable graph can be used as $\mathbf{L}_x$, we list here some reasonable choices, which are also used in our experiments. In the first option, GESSVDD-0, we have no data-dependent constraint, but $\mathbf{S}_x$ in (35) is replaced by an identity matrix $\mathbf{I}$, which corresponds to the orthogonality constraint. In the second option GESSVDD-I, we use $\mathbf{L}_x = \mathbf{I}$. The third option GESSVDD-PCA uses the PCA graph: $\mathbf{S}_x = \frac{1}{N}\mathbf{S}_t$.

While we only have samples from the positive class, it may include several clusters. To consider this option, we cluster the positive training samples using k-means and then define options GESSVDD-Sw and GESSVDD-Sb with $\mathbf{S}_x = \mathbf{S}_w$ and $\mathbf{S}_x = \mathbf{S}_b$, respectively. Here, $\mathbf{S}_w$ and $\mathbf{S}_b$ are solved as in (15) and (16), but c now refers to a cluster, not a class.

We also exploit the local geometric information by employing k-Nearest Neighbor (kNN) and setting

$$\mathbf{S}_x = \mathbf{S}_{kNN} = \mathbf{X}\Big(\mathbf{D}_{kNN} - \mathbf{A}_{kNN}\Big)\mathbf{X}^\intercal = \mathbf{X}\mathbf{L}_{kNN}\mathbf{X}^\intercal, \tag{46}$$

where $[\mathbf{A}]_{ij} = 1$, if $\mathbf{x}_i \in \mathcal{N}_j$ or $\mathbf{x}_j \in \mathcal{N}_i$ and 0, otherwise. $\mathcal{N}_i$ denotes the nearest neighbors of $\mathbf{x}_i$. This gives our last option denoted as GESSVDD-kNN.

Each of these options using different $\mathbf{S}_x$ can be solved using one of the update choices: gradient-based (GR), spectral ($\mathcal{S}$), or spectral regression-based (SR). Furthermore, in each case it is possible to either minimize or maximize the criterion in (35). To refer all these variants, we denote them as GESSVDD-0-GR-min, GESSVDD-0-GR-max, GESSVDD-0-$\mathcal{S}$-min and so on.

3.2. Framework analysis

Now we will get back to our main result, the general subspace learning framework for SVDD expressed as follows (repeated from (32)):

$$\mathrm{Tr}((\mathbf{Q}\mathbf{X}\mathbf{L}_x\mathbf{X}^\intercal\mathbf{Q}^\intercal)^{-1}\mathbf{Q}\mathbf{X}(\mathbb{A} - \boldsymbol{\alpha}\boldsymbol{\alpha}^\intercal)\mathbf{X}^\intercal\mathbf{Q}^\intercal), \tag{47}$$

where $\mathbf{L}_x$ can be used to enforce local/global data relations relevant for the task. Let us consider a graph with a weight matrix $[\mathbf{A}_\alpha]_{ij} = \alpha_i\alpha_j \; \forall i \neq j$ and $[\mathbf{A}_\alpha]_{ii} = 0$. With the constraint $\sum_{i=1}^N \alpha_i = 1$ (25), we get $[\mathbf{D}_\alpha]_{ii} = \sum_{j\neq i}[\mathbf{A}_\alpha]_{ij} = \sum_{j=1}^N \alpha_j\alpha_i - \alpha_i^2 = \alpha_i - \alpha_i^2$ and $\mathbf{L}_\alpha = \mathbf{D}_\alpha - \mathbf{A}_\alpha = \mathrm{diag}(\boldsymbol{\alpha}) - \boldsymbol{\alpha}\boldsymbol{\alpha}^\intercal = \mathbb{A} - \boldsymbol{\alpha}\boldsymbol{\alpha}^\intercal$. This shows that $\mathbb{A} - \boldsymbol{\alpha}\boldsymbol{\alpha}^\intercal$ is a graph Laplacian of a graph that connects the samples i and j with a weight $\alpha_i\alpha_j$. As α_i values are zero for any samples inside the hypersphere, the resulting graph has only connections between the support vectors and outliers.

We also see that the graph of $\mathbf{L}_\alpha$ has a strong similarity with the PCA graph. PCA maximizes the variance of the samples to their center $\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^N \mathbf{x}_i$, i.e.,

$$\begin{aligned}
\mathbf{S}_{pca} &= \frac{1}{N}\sum_{i=1}^N (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\intercal \\
&= \frac{1}{N}\sum_{i=1}^N (\mathbf{x}_i\mathbf{x}_i^\intercal - 2\mathbf{x}_i\boldsymbol{\mu}^\intercal + \boldsymbol{\mu}\boldsymbol{\mu}^\intercal) \\
&= \frac{1}{N}\sum_{i=1}^N (\mathbf{x}_i\mathbf{x}_i^\intercal) - 2\boldsymbol{\mu}\boldsymbol{\mu}^\intercal + \boldsymbol{\mu}\boldsymbol{\mu}^\intercal = \frac{1}{N}\sum_{i=1}^N (\mathbf{x}_i\mathbf{x}_i^\intercal) - \boldsymbol{\mu}\boldsymbol{\mu}^\intercal \\
&= \frac{1}{N}\mathbf{X}\mathbf{X}^\intercal - \frac{1}{N^2}\mathbf{X}\mathbf{1}\mathbf{1}^\intercal\mathbf{X}^\intercal = \frac{1}{N}\mathbf{X}(\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^\intercal)\mathbf{X}^\intercal \\
&= \mathbf{X}\mathbf{L}_{pca}\mathbf{X}^\intercal,
\end{aligned} \tag{48}$$

where $\mathbf{L}_{pca} = \mathbf{D}_{pca} - \mathbf{A}_{pca}$ and $[\mathbf{A}_{pca}]_{ij} = 1/N^2 \;\; \forall i \neq j$ and $[\mathbf{A}_{pca}]_{ii} = 0$. With an analogous derivation using the constraint $\sum_{i=1}^N \alpha_i = 1$, we see that $\mathbf{L}_\alpha$ represents the weighted variance of the support vectors and outliers to the center of SVDD defined as $\mathbf{a} = \sum_{i=1}^N \alpha_i\mathbf{x}_i$:

$$\begin{aligned}
\mathbf{S}_\alpha &= \sum_{i=1}^N (\mathbf{x}_i - \mathbf{a})(\mathbf{x}_i - \mathbf{a})^\intercal\alpha_i \\
&= \sum_{i=1}^N (\alpha_i\mathbf{x}_i\mathbf{x}_i^\intercal - 2\alpha_i\mathbf{x}_i\mathbf{a}^\intercal + \alpha_i\mathbf{a}\mathbf{a}^\intercal) \\
&= \sum_{i=1}^N (\alpha_i\mathbf{x}_i\mathbf{x}_i^\intercal) - 2\mathbf{a}\mathbf{a}^\intercal + \mathbf{a}\mathbf{a}^\intercal = \sum_{i=1}^N (\alpha_i\mathbf{x}_i\mathbf{x}_i^\intercal) - \mathbf{a}\mathbf{a}^\intercal \\
&= \mathbf{X}\mathrm{diag}(\boldsymbol{\alpha})\mathbf{X}^\intercal - \mathbf{X}\boldsymbol{\alpha}\boldsymbol{\alpha}^\intercal\mathbf{X}^\intercal = \mathbf{X}(\mathbb{A} - \boldsymbol{\alpha}\boldsymbol{\alpha}^\intercal)\mathbf{X}^\intercal \\
&= \mathbf{X}\mathbf{L}_\alpha\mathbf{X}^\intercal.
\end{aligned} \tag{49}$$

The main idea of PCA and SVDD along with graphs $\mathbf{L}_{pca}$ and $\mathbf{L}_\alpha$ are illustrated in Fig. 3.

By approximating the ratio trace in (47) with the corresponding trace ratio, we obtain a general subspace learning graph embedding framework with the graph preserving criterion

$$\begin{aligned}
\mathbf{Q}^* &= \underset{\mathrm{Tr}(\mathbf{Q}\mathbf{X}\mathbf{L}_x\mathbf{X}^\intercal\mathbf{Q}^\intercal)=m}{\arg\min} \sum_{i\neq j}(\mathbf{Q}\mathbf{x}_i - \mathbf{Q}\mathbf{x}_j)^2\alpha_i\alpha_j \\
&= \arg\min \frac{\mathrm{Tr}(\mathbf{Q}\mathbf{X}\mathbf{L}_\alpha\mathbf{X}^\intercal\mathbf{Q}^\intercal)}{\mathrm{Tr}(\mathbf{Q}\mathbf{X}\mathbf{L}_x\mathbf{X}^\intercal\mathbf{Q}^\intercal)}.
\end{aligned} \tag{50}$$

The criteria minimized in the previously proposed SSVDD [26,30] and ESSVDD [25] are special cases of the proposed framework and correspond to variants GESSVDD-0-GR-min and GESSVDD-I-GR-min. We conclude that SSVDD minimizes the weighted variance of the support vectors and outliers, while having an orthogonality constraint. ESSVDD also minimizes the weighted variance of the support vectors and outliers, while simultaneously maximizing the total scatter of the centered inputs.

Previously, SSVDD and ESSVDD used the gradient-based update of the projection vector. It should be noted that while the gradient-based approach moves only a single step toward the optimum of (28), the spectral and spectral regression-based updates proposed in Section 3.1 directly jump to the optimum. This may help the overall iterative GESSVDD process converge faster, but it may also introduce some instability, because the objectives of the iteration steps may be contradictory.

To summarize, the new framework in (47) places subspace learning for SVDD in the general graph embedding framework with a fixed data-dependent SVDD graph $\mathbf{L}_\alpha$, which resembles PCA on the support vectors and outliers, and an additional constraint graph $\mathbf{L}_x$, which allows to incorporate other meaningful data relationships to the subspace learning step. When the overall objective function in (47) is minimized, $\mathbf{L}_\alpha$ represents data relationships to be minimized and $\mathbf{L}_x$ represents data relationships to be maximized. In the earlier works, the overall objective function has been minimized via gradient-descent. However, the new framework hints that it can also make sense to reverse the objective and maximize instead of minimizing. Also this approach has been previously followed in the literature in [33], where kernel PCA was successfully applied for novelty detection.

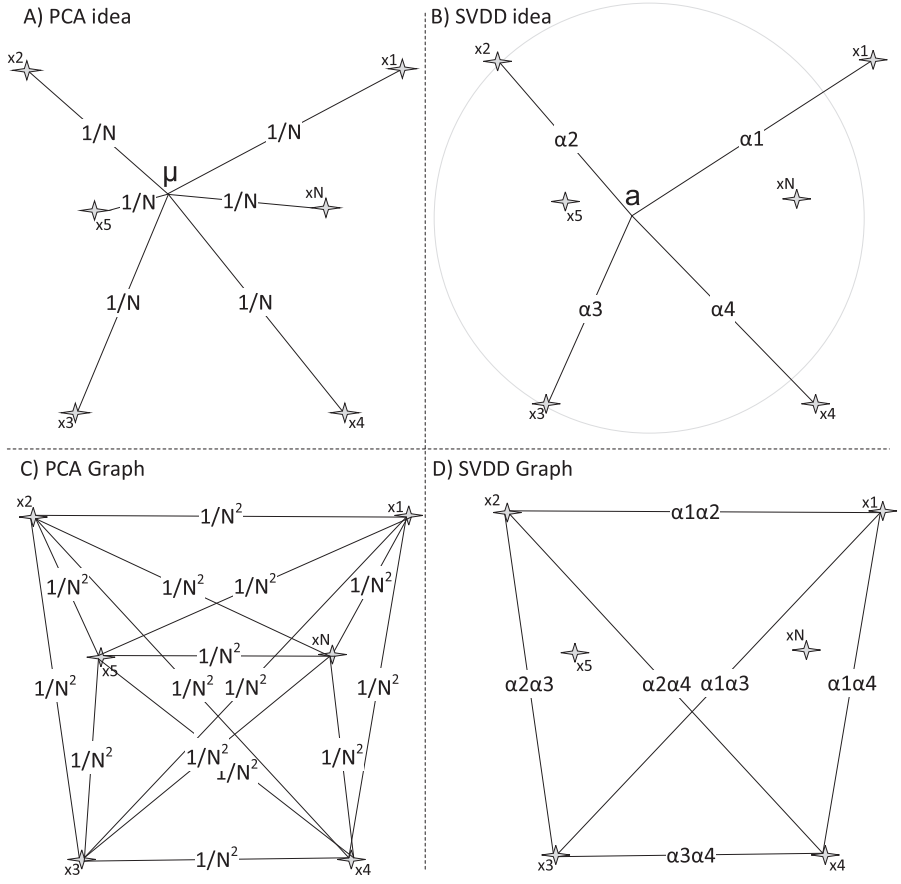

Fig. 3. A) PCA considers the (unweighted) variance of all the points from the center $\boldsymbol{\mu}$. B) SVDD considers weighted variance of support vectors and outliers from the SVDD center **a**. C) PCA graph is fully-connected with equal weights. D) SVDD graph is sparse (only the support vectors and outliers are connected) and has varying weights.

Intuitively, the original minimization of $\mathbf{L}_\alpha$ focuses on dimensions where the target class samples are the most similar, which indeed may help to discriminate the class from (unseen) other classes. On the other hand, from the similarity to PCA, we understand that these dimensions may be the dimensions that are not providing useful information in general (the corresponding PCA would discard them). Therefore, it is necessary to combine the criterion on $\mathbf{L}_\alpha$ with another criterion so that the combination can help to preserve the overall variance and minimize intra-class similarity simultaneously. In general, it may not be clear which criterion to minimize and which to maximize, but when considering the intra-cluster based graphs $\mathbf{L}_w$ and $\mathbf{L}_b$, an intuitive assumption is that within-cluster scatter $\mathbf{L}_w$ should be minimized (i.e., (47) maximized), while the between-cluster scatter $\mathbf{L}_b$ is more reasonable to be maximized (i.e., (47) minimized).

3.3. Complexity analysis

The proposed GESSVDD comprises three solutions: 1) gradient-based, 2) spectral, and 3) spectral regression-based updates. We first carry out the complexity analysis of the main algorithm (1), which contains the shared steps for all the updates, and then proceed to the steps different in each solution update. The following steps contribute to the overall complexity of the Algorithm 1:

1. Initializing of the projection matrix $\mathbf{Q}$ via PCA comprises two steps, i.e., computing the covariance matrix and then the eigenvalue decomposition. The complexity of these steps is $\mathcal{O}\big(ND \times \min(N, D)\big)$ and $\mathcal{O}\big(D^3\big)$, respectively.
2. Computing $\mathbf{S}_x = \mathbf{X}\mathbf{L}_x\mathbf{X}^\intercal$ for a given $\mathbf{L}_x$ has the complexity of $\mathcal{O}(DN^2 + ND^2)$.
3. Computing $\mathbf{S}_Q = \mathbf{Q}\mathbf{S}_x\mathbf{Q}^\intercal$ has the complexity of $\mathcal{O}(dD^2 + d^2D)$. Since, $D > d$, the complexity becomes $\mathcal{O}(dD^2)$.
4. Computing $\mathbf{S}_{inv}$ and the square-root of the matrix $\mathbf{S}_Q$ have the complexity of $\mathcal{O}(N^3)$.
5. SVDD has the complexity of $\mathcal{O}\big(N^3\big)$ for N data points [34].
6. The complexity of QR decomposition is $\mathcal{O}(dD^2)$ [35].

Dropping relatively lower computational costs and adding the rest, the complexity becomes $\mathcal{O}\big(N^3 + D^3\big)$. The total number of samples is assumed to be always greater than the dimensionality; hence the complexity becomes $\mathcal{O}\big(N^3\big)$. The complexity of each Sub-algorithm 2, 3, and 4 is $\mathcal{O}\big(N^3\big)$. We provide the details of

Sub-algorithm 1: Gradient-based update.

Input : $\mathbf{Q}$, $\mathbf{X}$, $\mathbf{S}_x$, $\mathbf{S}_{inv}$ $\mathbf{L}_\alpha$, η, *min/max* //Input from Algorithm 1
Output: $\mathbf{Q}$ //Return output to Algorithm 1

Compute $\mathbf{S}_\alpha = \mathbf{X}\mathbf{L}_\alpha\mathbf{X}^\intercal$;
Compute $\Delta L = 2\mathbf{S}_{inv}\mathbf{Q}\mathbf{S}_\alpha - 2\mathbf{S}_{inv}\mathbf{Q}\mathbf{S}_\alpha\mathbf{Q}^\intercal\mathbf{S}_{inv}\mathbf{Q}\mathbf{S}_x^\intercal$;
if *minimization*
Update $\mathbf{Q} \leftarrow \mathbf{Q} - \eta\Delta L$;
elseif *maximization*
Update $\mathbf{Q} \leftarrow \mathbf{Q} + \eta\Delta L$;

Sub-algorithm 2: Spectral update.

Input : $\mathbf{X}$, $\mathbf{S}_x$, $\mathbf{L}_\alpha$, *min/max* //Input from Algorithm 1
Output: $\mathbf{Q}$ //Return output to Algorithm 1

Compute $\mathbf{S}_\alpha = \mathbf{X}\mathbf{L}_\alpha\mathbf{X}^\intercal$;

Solve generalized eigenvalue problem$\mathbf{S}_\alpha\mathbf{q} = \upsilon\mathbf{S}_x\mathbf{q}$;

if *minimization*
Select the eigenvectors corresponding to d smallest positive eigenvalues as rows of $\mathbf{Q}$;
elseif *maximization*
Select the eigenvectors corresponding to d largest eigenvalues as rows of $\mathbf{Q}$;

Sub-algorithm 3: Spectral regression-based update.

Input : $\mathbf{X}$, $\mathbf{L}$, $\mathbf{L}_\alpha$, *min/max* //Input from Algorithm 1
Output: $\mathbf{Q}$ //Return output to Algorithm 1

Solve generalized eigenvalue problem:$\mathbf{L}_\alpha\mathbf{t} = \upsilon\mathbf{L}_x\mathbf{t}$;

if *minimization* then
Select the eigenvectors corresponding to d smallest positive eigenvalues as columns of $\mathbf{T}$;
elseif *maximization* then
Select the eigenvectors corresponding to d largest eigenvalues to as columns of $\mathbf{T}$;

Obtain $\mathbf{Q} = \mathbf{T}^\intercal\mathbf{X}^\intercal(\mathbf{X}\mathbf{X}^\intercal + \eta\mathbf{I})^{-1}$;

complexity analysis of Sub-algorithms 1, 2, and 3 in Sections *1.1. Complexity analysis of gradient-based update, 1.2. Complexity analysis of spectral-based update*, and *1.3. Complexity analysis of spectral regression-based update* respectively in the supplementary material. Adding the complexity of each Sub-algorithm to the main algorithm, the overall complexity remains at $\mathcal{O}(N^3)$, which is the same as for the original SVDD [34]. Moreover, in the non-linear case, the steps involved in NPT have the complexity of $\mathcal{O}(N^3)$; thus, the complexity in terms of the big $\mathcal{O}$ notation still stays as $\mathcal{O}(N^3)$.

4. Experiments

4.1. Datasets and experimental setup

To evaluate the proposed method's performance, we used nine different datasets. The datasets used in the experiments are Seeds, Qualitative bankruptcy, Somerville happiness, Liver, Iris, Ionosphere, Sonar, Heart (from UCI[1] machine learning repository) and MNIST [36] with original dimensionality D of 7, 6, 6, 6, 4, 34, 60, 13, and 784 respectively. MNIST has 10 classes, Seeds and Iris datasets are ternary, while the rest of the datasets are binary.

In Seeds dataset, the classes are named as Kama (S-K), Rosa (S-R), and Canadian (S-C) with 70 samples from each class. In Qualitative bankruptcy, the class labels are bankruptcy (QB-B) and non-bankruptcy (QB-N) with 107 and 143 samples, respectively. The Somerville happiness dataset contains 77 samples from the happy (SH-H) category and 66 from the unhappy (SH-U) category. Liver contain 145 samples from Disorder Present (DP) category and 200 samples from Disorder Absent (DA) category. Iris dataset contains 50 samples from each category of Setosa (I-S), Versicolor (S-VC), and Virginica (S-V). The Ionosphere dataset contains samples categorized as Bad (I-B) and Good (I-G). It contains 126 and 225 samples from bad and good categories, respectively. Sonar dataset has Rock (S-R) and Mines (S-M) as its two classes with 97 samples from Rock and 111 samples from Mines category. Heart dataset contain 139 samples from disease present and 164 samples from disease absent categories, respectively.

MNIST dataset contains 5923, 6742, 5958, 6131, 5842, 5421, 5918, 6265, 5851, 5949 samples in the training set for classes 0–9, respectively. In the test set, it contains 980, 1135, 1032, 1010, 982, 892, 958, 1028, 974, and 1009 from corresponding classes (0–9). In our experiments, we select 10% of the data from MNIST while keeping the representation of each class in train and test set similar to the original train and test split in the dataset.

We manually created a corrupted version of the heart dataset to report the impact of noise. We added the noise in the manner described in [37]. The corrupted data were created by adding pseudo-random values drawn from the standard normal distribution to the features. We bound the range of added noise for the corresponding attribute to the maximum and minimum value of each feature of the target class in the training set.

We converted these datasets into one-class classification datasets by considering a single class at a time as the positive class and the rest as the negative class. For MNIST, the train and test sets are given, so we used the original train and test splits for the experiments. We divided the rest of the datasets into train and test sets by considering 70% of data as training data and the remaining 30% as test data. We selected the 70-30 splits randomly by keeping the representation of each class similar to the original dataset. We performed the 70-30% selection five times; hence we repeated the experiment 5 times for a single scenario where each class is considered a positive class. Note that at this point, both the training and test sets contained samples from both positive and negative classes. We did not use the negative samples in the training set in optimizing the models but only to select the hyperparameters by using five-fold cross-validation within the training set. To this end, four of the folds (only positive items) at a time were used for optimizing the model, and the fifth fold (both positive and negative items) was used to evaluate the performance. Finally, we used the best-performing hyperparameter values to optimize the model with the entire training set (only positive items) and reported the performance over the test set. We used a similar setup for all the competing methods. During the five-fold cross-validation over the training set, we found the best hyperparameters from the following values: $C \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$, $\sigma \in \{10^{-1}, 10^0, 10^1, 10^2, 10^3\}$, $d \in \{1, 2, 3, 4, 5, 10, 20\}$, $\eta \in \{10^{-1}, 10^0, 10^1, 10^2, 10^3\}$. The number of iterations for all the iterative methods was set to 5.

As our evaluation metrics, we report Geometric Mean *Gmean*, True Positive Rate (*TPR*), True Negative Rate (*TNR*), False Positive Rate (*FPR*), and False Negative Rate (*FNR*), where $TPR = \frac{TP}{P}$, $TNR = \frac{TN}{N}$, $FPR = \frac{FP}{N}$, and $FNR = \frac{FN}{P}$. TP, TN, FP, FN, P, $\mathcal{N}$ denote true positives, true negatives, false positives, false negatives, and number of positive samples, and number of negative samples, respectively. We use *Gmean* as the main performance metric as it takes into account both *TPR* and *TNR*. We also report the standard deviations over the five data splittings.

For the proposed method, we consider all the variants introduced in Section 3.1.3: GESSVDD-0, GESSVDD-I, GESSVDD-PCA, GESSVDD-Sw, GESSVDD-Sb, and GESSVDD-kNN. For each, we consider all the alternative solutions (GR-gradient-based, $\mathcal{S}$-spectral, SR-spectral regression-based). The criterion in (35) is maximized and minimized in a separate set of experiments respectively for each variant and alternative solution. In order to construct the Laplacians $\mathbf{L}_w$ and $\mathbf{L}_b$, the number of clusters $\mathcal{C}$ was fixed to 5. Moreover, the numbers of neighbours for defining $\mathbf{L}_{kNN}$ was also fixed to 5.

We also carried out sensitivity analysis for the model for the range of hyperparameters. We followed the approach mentioned

[1] http://archive.ics.uci.edu/ml

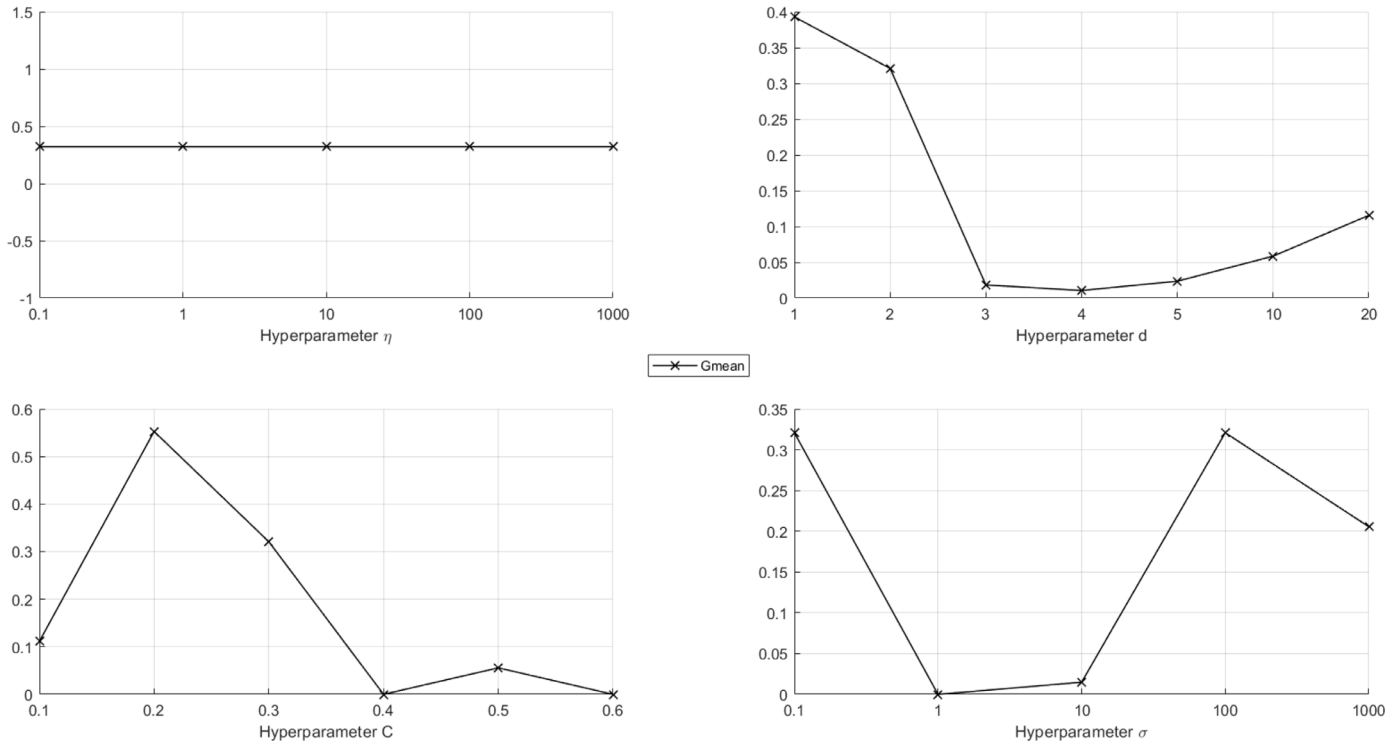

Fig. 4. Sensitivity Analysis for non-linear GESSVDD-kNN-SR-max trained over MNIST dataset with target class 0.

in [30] for sensitivity analysis. In order to analyze the sensitivity of the model for the corresponding hyperparameter, we fix other hyperparameters to their optimal values found over the training set and record the performance with all the hyperparameter values considered in the given range.

To evaluate whether the observed differences between different methods are statistically significant, we follow the recommendations of [38]. We perform Wilcoxon Sign-Ranks test over the average results for the nine datasets to evaluate the pair-wise differences between the methods. The test ranks the differences between each pair of classifiers ignoring the signs and uses the ranks to determine value T as described, e.g., in [38]. Finally, the T value is compared to a critical value which depends on the number of datasets. In our experiments, we used 9 datasets, which means that the null hypothesis can be rejected at 0.05 significance level if $T \leq 5$.

4.2. Experimental results and discussion

We report the results of the best performing linear and non-linear variants among the proposed variants compared against the previously proposed SSVDD [26] and ESSVDD [25], and the competing methods GESVM [23], GESVDD [23], OCSVM [21], SVDD [22], and ESVDD for all datasets in Table 1. In each experiment, a single class is used as a target class and the rest of the data as outliers. The average performance over each dataset is reported in the average (Av.) column. We report the average test results of different variants of the proposed framework over the five splittings of the Seeds, Qualitative bankruptcy, Somerville happiness, Iris, Ionosphere, and Sonar datasets in Table 2 for the non-linear data description, while the results over MNIST, Liver, and Heart datasets are reported in Section S2 of the supplementary material along with the results of all variants in the proposed framework in case of linear data description. We also provide *TPR*, *TNR*, *FPR*, and *FNR* results in S3 of the supplementary material. The corresponding standard deviations of *Gmean* over five splits are provided in Section S4 and Section S5 for linear and non-linear cases, respectively, in the supplementary material. Implementations of the proposed framework are available online in GitHub[2].

From the experimental results comparing different variants of GESSVDD, we observe that in both linear and non-linear methods, the gradient-based solution performs better than the spectral and spectral regression-based solutions in the majority of the cases. The spectral approaches are typically more unstable over iterations as discussed in Section 3.2. When comparing the minimization/maximization, we see that our claim that $\mathbf{L}_w$ should be used with maximization and $\mathbf{L}_b$ with minimization seems to be valid in most cases. Overall, minimization typically leads to better results. Moreover, the performance of kNN graph is better than that of other variants for both min and max cases and for both linear and non-linear methods.

Overall in linear methods, it is noted that employing the kNN graph for encoding geometric information in the subspace yields better results also compared to the competing methods in the majority of the cases. Linear GESSVDD-kNN-GR-min variant performs best over 5 and second-best over 2 out of 9 datasets. For non-linear methods, the different variants of GESSVDD have a more varying performance suggesting that finding a suitable graph for the task at hand may be more important. For comparisons, we report the results of a single variant GESSVDD-kNN-SR-max in the non-linear section of Table 1. It performs best over the Qualitative Bankruptcy and second-best over Seeds and Ionosphere datasets. For MNIST, we see that some other methods outperform the proposed variants by a clear margin. As the maximum dimensionality allowed for our proposed methods in our experiments is 20, whereas the original dimensionality of MNIST data is 784, we can conclude that the reduction in dimensionality is likely too dramatic for preserving the significant information.

[2] https://github.com/fahadsohrab/gessvdd

Table 1
Gmean results for linear and non-linear data description over different datasets, selected variants from the proposed framework vs. other one-class classification methods.

Dataset	Seeds				Qualitative bankruptcy			Somerville happiness			Liver		
Target class	S-K	S-R	S-C	Av.	QB-B	QB-N	Av.	SH-H	SH-U	Av.	DP	DA	Av.
Linear													
GESSVDD-kNN-GR-min	0.83	**0.94**	**0.95**	**0.91**	0.80	**0.46**	**0.63**	0.44	0.44	0.44	0.45	0.38	0.42
GESSVDD-I-GR-min (ESSVDD)	**0.87**	0.92	0.90	0.90	0.90	0.12	0.51	**0.51**	0.39	0.45	0.35	0.38	0.37
GESSVDD-0-GR-min (SSVDD)	0.85	0.93	**0.95**	**0.91**	0.90	0.17	0.53	0.49	0.43	0.46	0.32	0.34	0.33
ESVDD	0.79	0.87	0.87	0.84	**0.96**	0.19	0.58	0.42	0.41	0.41	0.35	**0.40**	0.38
SVDD	0.85	0.92	0.94	0.90	0.94	0.00	0.47	0.41	0.36	0.39	**0.50**	0.39	**0.45**
OCSVM	0.48	0.69	0.45	0.54	0.37	0.41	0.39	0.45	**0.53**	**0.49**	0.40	0.36	0.38
Non-Linear													
GESSVDD-kNN-SR-max	0.86	0.92	**0.96**	0.91	0.81	**0.71**	**0.76**	0.47	0.47	0.47	0.41	0.42	0.41
GESSVDD-I-GR-min (ESSVDD)	0.83	0.91	0.90	0.88	0.92	0.28	0.60	**0.59**	0.39	0.49	0.40	0.49	0.45
GESSVDD-0-GR-min (SSVDD)	**0.87**	**0.94**	0.94	**0.92**	0.94	0.46	0.70	0.47	0.35	0.41	0.37	0.39	0.38
ESVDD	0.81	0.88	0.87	0.85	0.00	0.00	0.00	0.00	0.31	0.16	0.43	0.54	0.49
SVDD	0.85	0.91	0.95	0.90	0.33	0.28	0.31	0.40	0.32	0.36	0.49	0.40	0.45
OCSVM	0.47	0.60	0.45	0.51	0.36	0.58	0.47	0.47	0.49	0.48	0.27	0.08	0.17
GESVDD-PCA	0.85	0.93	0.93	0.90	0.94	0.28	0.61	0.50	0.48	0.49	**0.51**	0.49	0.50
GESVDD-Sw	0.82	0.93	0.93	0.89	0.94	0.28	0.61	0.49	**0.50**	0.49	**0.51**	0.52	0.51
GESVDD-kNN	0.84	0.92	0.94	0.90	0.84	0.31	0.57	0.50	0.45	0.47	**0.51**	0.52	0.52
GESVM-PCA	0.85	0.90	0.93	0.89	**0.95**	0.26	0.60	0.52	0.48	**0.50**	0.50	**0.55**	0.52
GESVM-Sw	0.85	0.90	0.91	0.89	0.93	0.20	0.57	0.55	0.41	0.48	0.50	0.51	0.51
GESVM-kNN	0.84	0.90	0.90	0.88	0.92	0.20	0.56	0.55	0.51	0.53	**0.51**	0.55	**0.53**

Dataset	Iris				Ionosphere			Sonar			Heart		
Target class	I-S	I-VC	S-V	Av.	I-B	I-G	Av.	S-R	S-M	Av.	DP	DA	Av.
Linear													
GESSVDD-kNN-GR-min	**0.97**	0.89	**0.91**	**0.92**	0.42	**0.92**	**0.67**	**0.54**	0.57	**0.56**	0.54	0.61	0.58
GESSVDD-I-GR-min (ESSVDD)	0.93	0.82	0.89	0.88	0.36	0.90	0.63	0.52	**0.58**	0.55	0.53	**0.69**	**0.61**
GESSVDD-0-GR-min (SSVDD)	0.96	**0.91**	0.90	**0.92**	0.12	0.78	0.45	0.51	0.55	0.53	**0.59**	0.62	**0.61**
ESVDD	0.89	0.85	0.86	0.87	0.33	0.88	0.61	0.00	0.03	0.02	0.56	0.62	0.59
SVDD	0.92	0.90	0.89	0.91	0.02	0.86	0.44	0.52	0.56	0.54	0.46	0.35	0.41
OCSVM	0.58	0.50	0.46	0.51	**0.49**	0.51	0.50	0.48	0.45	0.46	0.57	0.63	0.60
Non-Linear													
GESSVDD-kNN-SR-max	**0.94**	0.87	0.83	0.88	**0.67**	0.86	0.76	0.52	0.47	0.49	0.42	0.43	0.42
GESSVDD-I-GR-min (ESSVDD)	**0.94**	0.88	0.89	0.90	0.64	0.89	**0.77**	0.54	0.55	0.54	0.38	0.37	0.37
GESSVDD-0-GR-min (SSVDD)	**0.94**	0.92	**0.90**	**0.92**	0.40	0.89	0.65	0.48	0.47	0.47	0.53	0.49	0.51
ESVDD	0.68	0.84	0.83	0.78	0.37	0.88	0.63	0.55	0.52	0.53	0.34	0.27	0.31
SVDD	0.92	0.92	0.88	0.90	0.21	0.85	0.53	0.53	0.59	0.56	0.53	0.55	0.54
OCSVM	0.56	0.26	0.55	0.46	0.52	0.47	0.49	0.47	0.55	0.51	0.20	0.23	0.21
GESVDD-PCA	0.83	0.92	0.89	0.88	0.38	0.88	0.63	**0.55**	**0.60**	0.57	0.68	**0.74**	**0.71**
GESVDD-Sw	0.89	0.87	0.90	0.89	0.36	0.90	0.63	0.53	0.54	0.54	0.68	0.73	0.70
GESVDD-kNN	0.83	0.91	0.89	0.88	0.34	0.89	0.62	0.54	**0.60**	**0.57**	**0.70**	0.72	0.71
GESVM-PCA	0.90	0.90	**0.90**	0.90	0.38	**0.91**	0.64	0.52	0.61	0.57	0.66	0.71	0.68
GESVM-Sw	0.89	**0.93**	0.88	0.90	0.45	0.90	0.67	0.54	0.59	0.57	0.67	0.70	0.68
GESVM-kNN	0.89	0.89	0.89	0.89	0.41	0.88	0.65	0.54	0.58	0.56	0.67	0.72	0.70

Dataset	MNIST											*Wilcoxon test*
Target class	0	1	2	3	4	5	6	7	8	9	Av.	*T*
Linear												
GESSVDD-kNN-GR-min	0.40	0.84	0.33	0.47	0.60	0.38	0.69	0.53	0.51	0.58	0.53	-
GESSVDD-I-GR-min (ESSVDD)	0.38	0.83	0.31	0.46	0.47	0.34	0.62	0.65	0.40	0.50	0.50	6.5
GESSVDD-0-GR-min (SSVDD)	0.41	0.81	0.29	0.39	0.45	0.31	0.57	0.52	0.40	0.44	0.46	9.0
ESVDD	0.00	0.81	0.00	0.00	0.00	0.00	0.06	0.22	0.00	0.16	0.13	**1.0**
SVDD	0.47	0.55	**0.51**	0.50	0.51	**0.52**	0.45	0.57	0.49	0.51	0.51	**5.0**
OCSVM	**0.57**	**0.92**	0.47	**0.52**	**0.64**	0.41	**0.73**	**0.74**	**0.53**	**0.63**	**0.62**	8.0
Non-Linear												
GESSVDD-kNN-SR-max	0.38	0.53	0.16	0.34	0.49	0.46	0.48	0.43	0.31	0.50	0.41	-
GESSVDD-I-GR-min (ESSVDD)	0.36	0.34	0.18	0.09	0.19	0.52	0.46	0.43	0.36	0.21	0.31	17.5
GESSVDD-0-GR-min (SSVDD)	0.60	0.34	0.48	0.39	0.43	0.49	0.43	0.35	0.44	0.17	0.41	15.5
ESVDD	0.54	0.19	0.34	0.14	0.39	0.52	0.42	0.32	0.17	0.36	0.34	**5.0**
SVDD	0.15	0.05	0.63	0.14	0.12	0.17	0.11	0.13	0.13	0.13	0.18	15.0
OCSVM	0.59	0.69	0.56	0.46	0.61	0.64	0.66	0.56	0.53	0.66	0.60	6.0
GESVDD-PCA	**0.92**	**0.96**	**0.75**	0.74	0.84	**0.73**	0.86	0.86	0.73	0.85	0.82	-15.5
GESVDD-Sw	**0.92**	**0.96**	**0.75**	0.74	0.84	0.72	0.00	0.85	0.71	0.85	0.74	-15.5
GESVDD-kNN	0.91	**0.96**	**0.75**	0.74	0.84	0.72	0.86	0.86	0.73	0.85	0.82	-17.5
GESVM-PCA	0.90	0.95	**0.75**	0.74	**0.87**	0.71	0.89	**0.86**	**0.76**	**0.86**	**0.83**	-14.5
GESVM-Sw	0.90	0.95	**0.75**	0.74	0.85	0.66	0.87	0.84	0.75	0.85	0.82	-14.5
GESVM-kNN	0.90	0.95	0.74	**0.76**	**0.87**	0.71	**0.89**	0.85	0.73	0.85	0.82	-14.0

Table 2
Gmean results for non-linear data description in the proposed framework.

Dataset	Seeds				Qualitative bankruptcy			Somerville happiness		
Target class	S-K	S-R	S-C	Av.	QB-B	QB-N	Av.	SH-H	SH-U	Av.
GESSVDD-Sb-$\mathcal{S}$-max	0.75	0.91	0.88	0.85	0.58	0.47	0.52	0.37	0.33	0.35
GESSVDD-Sb-GR-max	0.83	0.72	0.89	0.81	0.82	0.36	0.59	0.41	0.40	0.41
GESSVDD-Sb-SR-max	0.77	0.81	0.93	0.83	0.56	0.29	0.43	0.44	0.30	0.37
GESSVDD-Sb-$\mathcal{S}$-min	0.79	0.90	0.84	0.84	0.61	0.42	0.52	0.43	0.39	0.41
GESSVDD-Sb-GR-min	0.83	0.61	0.91	0.78	0.80	0.50	0.65	0.50	0.42	0.46
GESSVDD-Sb-SR-min	0.72	0.86	0.92	0.83	0.56	0.47	0.52	0.45	0.30	0.37
GESSVDD-Sw-$\mathcal{S}$-max	0.85	0.89	0.92	0.89	0.84	0.25	0.55	0.53	0.37	0.45
GESSVDD-Sw-GR-max	0.88	0.86	0.91	0.88	0.81	0.12	0.46	0.47	0.43	0.45
GESSVDD-Sw-SR-max	0.82	0.89	0.89	0.87	0.85	0.53	0.69	0.51	0.36	0.44
GESSVDD-Sw-$\mathcal{S}$-min	0.78	0.89	0.92	0.86	0.86	0.33	0.60	0.55	**0.50**	**0.52**
GESSVDD-Sw-GR-min	**0.89**	0.94	0.92	**0.92**	0.77	0.03	0.40	0.49	0.42	0.45
GESSVDD-Sw-SR-min	0.81	0.87	0.91	0.87	0.93	0.71	**0.82**	0.55	0.44	0.49
GESSVDD-kNN-$\mathcal{S}$-max	0.87	0.90	0.90	0.89	0.73	0.62	0.68	0.51	0.34	0.42
GESSVDD-kNN-GR-max	0.82	0.91	0.89	0.87	0.88	0.28	0.58	0.55	0.41	0.48
GESSVDD-kNN-SR-max	0.86	0.92	**0.96**	0.91	0.81	0.71	0.76	0.47	0.47	0.47
GESSVDD-kNN-$\mathcal{S}$-min	0.87	0.88	0.94	0.89	0.80	**0.78**	0.79	0.49	0.43	0.46
GESSVDD-kNN-GR-min	0.84	0.94	0.91	0.90	0.85	0.38	0.61	**0.60**	0.39	0.49
GESSVDD-kNN-SR-min	0.87	0.89	0.94	0.90	0.76	0.75	0.76	0.46	0.38	0.42
GESSVDD-PCA-$\mathcal{S}$-max	0.83	0.91	0.94	0.89	0.60	0.67	0.63	0.51	0.46	0.48
GESSVDD-PCA-GR-max	0.78	0.90	0.90	0.86	0.90	0.14	0.52	0.56	0.37	0.46
GESSVDD-PCA-SR-max	0.83	0.74	0.94	0.84	0.90	0.40	0.65	0.48	0.46	0.47
GESSVDD-PCA-$\mathcal{S}$-min	0.85	0.94	0.94	0.91	0.85	0.48	0.67	0.55	0.38	0.47
GESSVDD-PCA-GR-min	0.86	0.86	0.94	0.89	0.93	0.17	0.55	0.53	0.40	0.46
GESSVDD-PCA-SR-min	0.84	0.90	0.94	0.89	0.93	0.61	0.77	0.51	0.42	0.47
GESSVDD-I-$\mathcal{S}$-max	0.85	0.93	0.76	0.84	0.83	0.39	0.61	0.43	0.37	0.40
GESSVDD-I-GR-max	0.83	0.91	0.91	0.88	0.91	0.13	0.52	0.52	0.41	0.47
GESSVDD-I-SR-max	0.85	0.93	0.94	0.91	0.86	0.54	0.70	0.49	0.44	0.46
GESSVDD-I-$\mathcal{S}$-min	0.85	0.94	0.93	0.91	0.85	0.42	0.64	0.47	0.42	0.44
GESSVDD-I-GR-min (ESSVDD)	0.83	0.91	0.90	0.88	0.92	0.28	0.60	0.59	0.39	0.49
GESSVDD-I-SR-min	0.85	**0.95**	0.93	0.91	0.84	0.49	0.67	0.54	0.38	0.46
GESSVDD-0-$\mathcal{S}$-max	0.85	0.92	0.93	0.90	0.70	0.44	0.57	0.39	0.44	0.41
GESSVDD-0-GR-max	0.85	0.93	0.94	0.90	0.93	0.47	0.70	0.37	0.46	0.42
GESSVDD-0-$\mathcal{S}$-min	0.86	0.91	0.92	0.89	0.70	0.47	0.59	0.38	0.38	0.38
GESSVDD-0-GR-min (SSVDD)	0.87	0.94	0.94	**0.92**	**0.94**	0.46	0.70	0.47	0.35	0.41

Dataset	Iris				Ionosphere			Sonar		
Target class	I-S	S-VC	S-V	Av.	I-B	I-G	Av.	S-R	S-M	Av.
GESSVDD-Sb-$\mathcal{S}$-max	0.94	0.88	0.77	0.86	0.57	0.63	0.60	0.42	0.38	0.40
GESSVDD-Sb-GR-max	0.95	0.83	0.90	0.89	0.42	0.87	0.64	0.50	0.40	0.45
GESSVDD-Sb-SR-max	0.80	0.85	0.85	0.83	0.53	0.42	0.47	0.40	0.39	0.40
GESSVDD-Sb-$\mathcal{S}$-min	0.93	0.90	0.87	0.90	0.46	0.52	0.49	0.42	0.47	0.45
GESSVDD-Sb-GR-min	0.95	0.86	0.90	0.90	0.30	0.87	0.59	0.50	0.49	0.50
GESSVDD-Sb-SR-min	0.82	0.82	0.74	0.79	0.51	0.42	0.46	0.40	0.34	0.37
GESSVDD-Sw-$\mathcal{S}$-max	0.85	0.91	0.84	0.87	0.49	0.86	0.67	0.54	0.38	0.46
GESSVDD-Sw-GR-max	0.94	0.90	0.89	0.91	0.48	0.89	0.68	0.52	0.46	0.49
GESSVDD-Sw-SR-max	0.97	0.86	0.85	0.89	0.35	0.87	0.61	0.37	0.50	0.44
GESSVDD-Sw-$\mathcal{S}$-min	0.74	0.89	0.85	0.83	0.46	0.86	0.66	0.51	0.50	0.51
GESSVDD-Sw-GR-min	0.95	0.86	0.83	0.88	0.61	0.87	0.74	0.50	0.44	0.47
GESSVDD-Sw-SR-min	0.97	0.86	0.86	0.89	0.32	0.87	0.59	0.37	0.41	0.39
GESSVDD-kNN-$\mathcal{S}$-max	0.94	0.88	0.83	0.88	**0.67**	0.87	0.77	0.51	0.48	0.50
GESSVDD-kNN-GR-max	0.94	0.92	0.88	0.91	0.65	0.90	**0.78**	0.54	0.49	0.51
GESSVDD-kNN-SR-max	0.94	0.87	0.83	0.88	**0.67**	0.86	0.76	0.52	0.47	0.49
GESSVDD-kNN-$\mathcal{S}$-min	0.94	0.88	0.83	0.89	0.38	0.87	0.62	**0.58**	0.43	0.50
GESSVDD-kNN-GR-min	0.92	0.88	**0.91**	0.90	0.59	**0.92**	0.75	0.54	**0.59**	**0.57**
GESSVDD-kNN-SR-min	0.94	0.87	0.82	0.88	0.49	0.86	0.67	0.54	0.56	0.55
GESSVDD-PCA-$\mathcal{S}$-max	0.94	**0.95**	0.87	**0.92**	0.37	0.87	0.62	0.42	0.36	0.39
GESSVDD-PCA-GR-max	0.92	0.87	0.85	0.88	0.57	0.89	0.73	0.54	0.43	0.48
GESSVDD-PCA-SR-max	0.94	0.92	0.88	0.91	0.34	0.84	0.59	0.29	0.52	0.41
GESSVDD-PCA-$\mathcal{S}$-min	0.94	0.92	0.80	0.89	0.37	0.87	0.62	0.50	0.51	0.50
GESSVDD-PCA-GR-min	0.92	0.73	0.85	0.83	0.59	0.91	0.75	0.53	0.53	0.53
GESSVDD-PCA-SR-min	0.94	0.92	0.83	0.90	0.30	0.87	0.58	0.56	0.49	0.53
GESSVDD-I-$\mathcal{S}$-max	**0.98**	0.92	0.84	**0.92**	0.25	0.88	0.56	0.48	0.36	0.42
GESSVDD-I-GR-max	0.95	0.86	0.88	0.90	0.63	0.88	0.76	0.56	0.54	0.55
GESSVDD-I-SR-max	**0.98**	0.93	0.84	**0.92**	0.24	0.88	0.56	0.49	0.51	0.50
GESSVDD-I-$\mathcal{S}$-min	0.95	0.93	0.84	0.91	0.22	0.88	0.55	0.50	0.52	0.51
GESSVDD-I-GR-min (ESSVDD)	0.94	0.88	0.89	0.90	0.64	0.89	0.77	0.54	0.55	0.54
GESSVDD-I-SR-min	0.95	0.93	0.85	0.91	0.29	0.88	0.58	0.53	0.54	0.53
GESSVDD-0-$\mathcal{S}$-max	0.94	0.91	0.90	0.91	0.61	0.63	0.62	0.47	0.58	0.52
GESSVDD-0-GR-max	0.96	0.91	0.88	**0.92**	0.44	0.82	0.63	0.49	0.54	0.51
GESSVDD-0-$\mathcal{S}$-min	0.95	0.88	0.90	0.91	0.41	0.73	0.57	0.51	0.40	0.45
GESSVDD-0-GR-min (SSVDD)	0.94	0.92	0.90	**0.92**	0.40	0.89	0.65	0.48	0.47	0.47

Table 3
Gmean results for linear and non-linear data description over manually created corrupted versions of heart dataset, selected variants from the proposed framework vs. other one-class classification methods.

Dataset	Heart Clean train set Corrupted test set			Heart Corrupted train set Clean test set			Heart Corrupted train set Corrupted test set		
Target class	DP	DA	Av.	DP	DA	Av.	DP	DA	Av.
Linear									
GESSVDD-Sb-GR-max	0.17	0.22	0.20	0.24	**0.36**	0.30	0.36	0.30	0.33
GESSVDD-kNN-GR-min	0.09	0.00	0.04	0.32	0.08	0.20	0.35	0.44	0.39
GESSVDD-I-GR-min (ESSVDD)	0.00	0.00	0.00	0.00	0.00	0.00	0.35	0.39	0.37
GESSVDD-0-GR-min (SSVDD)	0.00	0.17	0.09	0.00	0.00	0.00	0.36	0.40	0.38
ESVDD	0.00	0.00	0.00	0.00	0.00	0.00	0.43	0.49	0.46
SVDD	**0.38**	**0.41**	**0.39**	**0.42**	0.27	**0.35**	**0.48**	**0.51**	**0.49**
OCSVM	0.00	0.00	0.00	0.00	0.00	0.00	0.34	0.41	0.38
Non-Linear									
GESSVDD-Sb-GR-max	**0.35**	0.23	**0.29**	**0.33**	0.31	**0.32**	0.27	0.45	0.36
GESSVDD-kNN-SR-max	0.04	0.03	0.03	0.00	0.00	0.00	0.47	0.40	0.44
GESSVDD-I-GR-min (ESSVDD)	0.37	0.15	0.26	0.00	0.11	0.06	0.38	0.46	0.42
GESSVDD-0-GR-min (SSVDD)	0.12	**0.24**	0.18	0.00	0.09	0.04	0.37	0.47	0.42
ESVDD	0.00	0.00	0.00	0.00	0.03	0.02	0.49	0.52	0.51
SVDD	0.07	0.15	0.11	0.22	0.07	0.15	0.47	0.45	0.46
OCSVM	0.00	0.00	0.00	0.00	0.07	0.03	**0.50**	0.49	0.50
GESVDD-PCA	0.00	0.00	0.00	0.00	0.00	0.00	0.51	**0.53**	**0.52**
GESVDD-Sw	0.00	0.00	0.00	0.00	0.00	0.00	0.53	0.52	**0.52**
GESVDD-kNN	0.00	0.00	0.00	0.00	0.00	0.00	0.48	0.52	0.50
GESVM-PCA	0.00	0.00	0.00	0.00	0.00	0.00	0.49	0.49	0.49
GESVM-Sw	0.00	0.00	0.00	0.00	0.05	0.03	**0.50**	0.48	0.49
GESVM-kNN	0.00	0.00	0.00	0.00	0.03	0.02	0.49	0.50	0.49

We applied Wilcoxon Sign-Ranks separately for linear and non-linear methods. We compared all other linear methods in Table 1 against our proposed GESSVDD-kNN-GR-min variant and all other non-linear methods Table 1 against our proposed GESSVDD-kNN-SR-max variant. We give the T-values in Table 1 and bold the values if they show that the difference between the methods is statistically significant at 0.05 significance level. Negative values indicate that the other method was performing better than our proposed variant GESSVDD-kNN-GR-min or GESSVDD-kNN-SR-max. We see that GESSVDD-kNN-GR-min outperforms ESVDD and SVDD in a statistically significant manner for linear data description and GESSVDD-kNN-SR-max outperforms ESVDD in a statistically significant manner for non-linear data description. All other differences are statistically insignificant. However, it should be noted that for individual datasets the differences in both ways can be still significant due to different reasons, such as our proposed variant failing with MNIST due to the drastic dimensionality reduction, and it cannot be concluded that the selection of the method is insignificant.

In evaluating the effect of added noise on the features of the heart dataset, it can be noticed that GESSVDD-Sb-GR-max performs second-best in the linear case when only the train or test set is corrupted. In the non-linear case of adding noise to either train or test set, GESSVDD-Sb-GR-max performs best on average. While the competing methods perform better than the proposed methods when both train and test datasets are corrupted, the competing methods underperform severely if the only train or test set is corrupted. There is not a single case where the proposed method would severely underperform. We report the performance of the selected variants of our method along with the competing methods in Table 3. We provide the *Gmean* results for all proposed variants over the Heart dataset and its manually created corrupted versions in Section S2 and *TPR*, *TNR FPR*, and *FNR* results in Section S3 of the supplementary material.

We carried out a sensitivity analysis of different hyperparameters. Fig. 4 shows the sensitivity plot for non-linear GESSVDD-kNN-SR-max trained over MNIST dataset with target class 0. For all other variants, we provide the plots of sensitivity analysis in Section S6 of the supplementary material. We observe that the performance of GESSVDD-kNN-SR-max is not sensitive to the hyperparameter η. In the case of increasing the value of hyperparameter d, a sudden drop and then a steady rise in the performance is observed over the range of values. We also notice the poor performance of the model at higher values of hyperparameter C; moreover, a varying performance is noticed at different values for hyperparameter σ.

5. Conclusion

In this paper, we formulated subspace learning for one-class classification in the graph embedding framework and discussed the novel insights obtained from this formulation. In particular, we showed that subspace learning for SVDD applies a weighted PCA over the support vectors and outliers to define the projection matrix and we discussed how this information can be combined with other data relationships in the optimization process via an adaptable graph. We also formulated a novel Graph-Embedded Subspace Support Vector Data Description with gradient-based, spectral, and spectral regression-based solutions and different adaptable graphs. We reported the experimental results over nine different datasets by considering each class of a dataset as a target class at a time. The results showed that the proposed framework with the kNN graph as the adaptable graph had the best overall performance, while the gradient-based solution was more stable than the spectral and spectral regression-based solutions.

While the proposed framework showed promising results over different datasets and can be applied on different domain applications, there are some limitations that can be taken into account in the future. The methods exploit only a single Laplacian $\mathbf{L}_x$ to enforce local/global data relations relevant to the task. This can be enhanced by exploiting multiple graphs by combining the geometric data relationships using a weight parameter.

In the future, we plan to extend the proposed methods in the framework by investigating other kernel types in the non-linear case. The proposed framework can also be extended to multi-modal one-class classification, where data is projected from multiple modalities to a joint subspace.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work has been supported by NSF IUCRC CVDI, project AMALIA funded by Business Finland and DSB, as well as projects Mad@work and Stroke-Data funded by Haltian. The work of Jenni Raitoharju was supported by Academy of Finland project 324475.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.patcog.2022.108999

References

[1] N. Vaswani, T. Bouwmans, S. Javed, P. Narayanamurthy, Robust subspace learning: robust pca, robust subspace tracking, and robust subspace recovery, IEEE Signal Process. Mag. 35 (4) (2018) 32–55.
[2] X.-L. Xu, C.-X. Ren, R.-C. Wu, H. Yan, Sliced inverse regression with adaptive spectral sparsity for dimension reduction, IEEE Trans. Cybern. 47 (3) (2017) 759–771, doi:10.1109/TCYB.2016.2526630.
[3] J. Guo, X. Li, Based on statistics of the gradients the feature matching algorithm, in: International Workshop on Education Technology and Computer Science, volume 2, 2009, pp. 983–987.
[4] E. Rodriguez-Martinez, T. Mu, J.Y. Goulermas, Sequential projection pursuit with kernel matrix update and symbolic model selection, IEEE Trans. Cybern. 44 (12) (2014) 2458–2469.
[5] X. He, C. Zhang, L. Zhang, X. Li, A-Optimal projection for image representation, IEEE Trans. Pattern Anal. Mach. Intell. 38 (5) (2015) 1009–1015.
[6] L. Xu, J. Raitoharju, A. Iosifidis, M. Gabbouj, Saliency-based weighted multi-label linear discriminant analysis, IEEE Trans. Cybern.(early access) (2021).
[7] Y. Lim, J. Kwon, H.-S. Oh, Principal component analysis in the wavelet domain, Pattern Recognit. (2021) 108096.
[8] R. Sheikh, M. Patel, A. Sinhal, Recognizing mnist handwritten data set using pca and lda, in: International Conference on Artificial Intelligence: Advances and Applications, 2020, pp. 169–177.
[9] D. Cai, X. He, J. Han, Srda: an efficient algorithm for large-scale discriminant analysis, IEEE Trans. Knowl. Data Eng. 20 (1) (2007) 1–12.
[10] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, C.A. Raffel, Mixmatch: A holistic approach to semi-supervised learning, in: Advances in Neural Information Processing Systems, 2019, pp. 5049–5059.
[11] T. Kefi-Fatteh, R. Ksantini, M.-B. Kaâniche, A. Bouhoula, A novel incremental one-class support vector machine based on low variance direction, Pattern Recognit. 91 (2019) 308–321.
[12] O.U. Lenz, D. Peralta, C. Cornelis, Average localised proximity: a new data descriptor with good default one-class classification performance, Pattern Recognit. 118 (2021) 107991.
[13] S. Alam, S.K. Sonbhadra, S. Agarwal, P. Nagabhushan, One-class support vector classifiers: a survey, Knowl. Based Syst. (2020) 105754.
[14] L.M. Manevitz, M. Yousef, One-class svms for document classification, J. Mach. Learn. Res. 2 (Dec) (2001) 139–154.
[15] G. Cohen, H. Sax, A. Geissbuhler, et al., Novelty detection using one-class parzen density estimator. an application to surveillance of nosocomial infections, in: Mie, 2008, pp. 21–26.
[16] M. Hejazi, Y.P. Singh, One-class support vector machines approach to anomaly detection, Appl. Artif. Intell. 27 (5) (2013) 351–366.
[17] F. Sohrab, J. Raitoharju, Boosting rare benthic macroinvertebrates taxa identification with one-class classification, in: IEEE Symposium Series on Computational Intelligence, 2020, pp. 928–933.
[18] W. Khreich, B. Khosravifar, A. Hamou-Lhadj, C. Talhi, An anomaly detection system based on variable n-gram features and one-class svm, Inf. Softw. Technol. 91 (2017) 186–197.
[19] L. Yin, H. Wang, W. Fan, Active learning based support vector data description method for robust novelty detection, Knowl. Based Syst. 153 (2018) 40–52.
[20] H.-J. Xing, Y.-J. Liu, Z.-C. He, Robust sparse coding for one-class classification based on correntropy and logarithmic penalty function, Pattern Recognit. 111 (2021) 107685.
[21] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, Sv estimation of a distribution's support, Adv. Neural Inf. Process. Syst. 12 (1999).
[22] D.M. Tax, R.P. Duin, Support vector data description, Mach. Learn. 54 (1) (2004) 45–66.
[23] V. Mygdalis, A. Iosifidis, A. Tefas, I. Pitas, Graph embedded one-class classifiers for media data classification, Pattern Recognit. 60 (2016) 585–595.
[24] M. Turkoz, S. Kim, Y. Son, M.K. Jeong, E.A. Elsayed, Generalized support vector data description for anomaly detection, Pattern Recognit. 100 (2020) 107119.
[25] F. Sohrab, J. Raitoharju, A. Iosifidis, M. Gabbouj, Ellipsoidal subspace support vector data description, IEEE Access 8 (2020) 122013–122025.
[26] F. Sohrab, J. Raitoharju, M. Gabbouj, A. Iosifidis, Subspace support vector data description, in: International Conference on Pattern Recognition, 2018, pp. 722–727.
[27] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, S. Lin, Graph embedding and extensions: a general framework for dimensionality reduction, IEEE Trans. Pattern Anal. Mach. Intell. 29 (1) (2006) 40–51.
[28] D. Cai, X. He, J. Han, Spectral regression for dimensionality reduction, Technical Report, Computer Science Department, UIUC, 2007.
[29] T. Hastie, R. Tibshirani, J. Friedman, The elements of statistical learning: Data mining, inference, and prediction, Springer Science & Business Media, 2009.
[30] F. Sohrab, J. Raitoharju, A. Iosifidis, M. Gabbouj, Multimodal subspace support vector data description, Pattern Recognit. 110 (2021) 107648.
[31] K.B. Petersen, M.S. Pedersen, The matrix cookbook, 2012. Version 20121115, https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf.
[32] N. Kwak, Nonlinear projection trick in kernel methods: an alternative to the kernel trick, IEEE Trans. Neural Netw. Learn. Syst. 24 (12) (2013) 2113–2119.
[33] H. Hoffmann, Kernel pca for novelty detection, Pattern Recognit. 40 (3) (2007) 863–874.
[34] S. Zheng, Smoothly approximated support vector domain description, Pattern Recognit. 49 (2016) 55–64.
[35] A. Sharma, K.K. Paliwal, S. Imoto, S. Miyano, Principal component analysis using QR decomposition, Int. J. Mach. Learn. Cybern. 4 (2013).
[36] L. Deng, The mnist database of handwritten digit images for machine learning research, IEEE Signal Process. Mag. 29 (6) (2012) 141–142.
[37] X. Zhu, X. Wu, Class noise vs. attribute noise: a quantitative study, Artif. Intell. Rev. 22 (3) (2004) 177–210.
[38] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.

Fahad Sohrab is a Postdoctoral Research Fellow at the Department of Computing Sciences, Tampere University, Finland. Dr. Fahad Sohrab received his MS degree from Sabanci University, Istanbul Turkey, in 2016 and his PhD degree from Tampere University, Finland in 2022. His research interests include machine learning, pattern recognition, and anomaly detection.

Alexandros Iosifidis received his PhD degree in Informatics from the Aristotle University of Thessaloniki in 2014. He is a Professor in Machine Learning and Computational Intelligence at Aarhus University, Denmark. His research interests include statistical machine learning and artificial neural networks with applications in Computer Vision, Finance and graph analysis problems.

Moncef Gabbouj received his MS and PhD degrees in electrical engineering from Purdue University, in 1986 and 1989, respectively. Dr. Gabbouj is Professor of Signal Processing at the Department of Computing Sciences, Tampere University, Finland. His research interests include Big Data analytics, multimedia analysis, artificial intelligence, machine learning, and pattern recognition.

Jenni Raitoharju received her PhD in Information Technology from Tampere University of Technology in 2017. She is an Assistant Professor of Signal Processing at University of Jyväskylä, Finland and a Senior Research Scientist at the Finnish Environment Institute, Finland. Her research interests include machine learning and pattern recognition methods along with applications in biomonitoring and autonomous systems.