2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Newton Method-based Subspace Support Vector Data Description

Fahad Sohrab
Faculty of Information Technology
and Communication Sciences
Tampere University
Tampere, Finland
fahad.sohrab@tuni.fi

Firas Laakom
Faculty of Information Technology
and Communication Sciences
Tampere University
Tampere, Finland
firas.laakom@tuni.fi

Moncef Gabbouj
Faculty of Information Technology
and Communication Sciences
Tampere University
Tampere, Finland
moncef.gabbouj@tuni.fi

***Abstract*—In this paper, we present an adaptation of Newton's method for the optimization of Subspace Support Vector Data Description (S-SVDD). The objective of S-SVDD is to map the original data to a subspace optimized for one-class classification, and the iterative optimization process of data mapping and description in S-SVDD relies on gradient descent. However, gradient descent only utilizes first-order information, which may lead to suboptimal results. To address this limitation, we leverage Newton's method to enhance data mapping and data description for an improved optimization of subspace learning-based one-class classification. By incorporating this auxiliary information, Newton's method offers a more efficient strategy for subspace learning in one-class classification as compared to gradient-based optimization. The paper discusses the limitations of gradient descent and the advantages of using Newton's method in subspace learning for one-class classification tasks. We provide both linear and nonlinear formulations of Newton's method-based optimization for S-SVDD. In our experiments, we explored both the minimization and maximization strategies of the objective. The results demonstrate that the proposed optimization strategy outperforms the gradient-based S-SVDD in most cases.**

***Index Terms*—One-class Classification, Support Vector Data Description, Subspace Learning**

I. INTRODUCTION

Traditional machine learning binary classification tasks focus on developing models that accurately classify samples into two categories. However, when faced with imbalanced datasets, where one category has a significantly larger number of samples than the other, machine learning models tend to exhibit bias towards the majority category. This bias can lead to suboptimal performance. To address this issue, one-class classification techniques can be used. These techniques involve training a model using only data from one category rather than both categories. This can help to mitigate bias and improve the model's ability to classify samples from the minority category.

One-Class Classification (OCC) approaches are a set of unsupervised learning techniques, typically carried out by using instances in the target class. The trained model is used to distinguish the target class from the rest of all classes, referred to as outliers. Considerable research has been carried out over the last few decades on learning algorithms for OCC, and these techniques have been used for various applications. In [1], OCC is used for automated leaf disease detection in different crop species. In [2], OCC technique is used to detect tumors in brain CT images. In [3], a fault detection technique for grid-connected photovoltaic Inverters based on OCC is proposed. In [4], the capability of OCC to complement deep CNN-based taxa identification by indicating samples potentially belonging to the rare classes of interest for human inspection is analyzed. In [5], uni-modal and multi-modal one-class classification techniques are examined for Early Myocardial Infarction Detection. More recently, in [6], OCC is used for hyperspectral image analysis.

OCC methods can be broadly divided into support vector-based and non-support vector-based approaches [7]. The support vector-based methods identify the so-called support vectors in the training set and are used to infer the decision boundary. The non-support vector-based approaches, such as density estimation and reconstruction-based methods, are used when the data's distribution or data generation process is accurately known. While most of the traditional algorithms for OCC such as One-class Support Vector Machine (OC-SVM) [8] and Support Vector Data Description (SVDD) [9] operate for data points in the original feature space, there has been a rising trend of algorithms where the method maps the original data to a lower dimensional feature space more suitable for one-class classification.

Subspace Support vector data description (S-SVDD) [10] is one such example of OCC methods where the model maps the data to a subspace optimized for one-class classification. The iterative optimization process of data mapping and the data description relies on gradient descent (GD). However, GD only relies on first-order information to find its solution, which can be sub-optimal [11], [12]. In this work, we propose a novel variant of S-SVDD, where the data mapping and parameters are iteratively optimized using Newton's method rather than gradient descent. In addition to first-order information, Newton's method uses second-order information, i.e., Hessian matrix, to optimize the objective.

Recently, there has been an interest in using second-order optimization methods in general and Newton's method in particular in the machine learning context [13]–[16], as these methods have several desired theoretical properties [12] and have shown superior empirical performance in several tasks [13], [17]–[20]. One major limitation of GD is sensitivity

978-0-7381-4408-5/23/$31.00 ©2023 IEEE

to scaling and linear transformations, which can lead to being trapped in trivial solutions [12]. On the other hand, Newton's method does not have this limitation and is affine invariant. Furthermore, Newton's method has, in general, faster convergence rates compared to GD and yields superior performance in practice [13], [16]–[18], [21].

Moreover, as the S-SVDD optimization problem involves primarily quadratic terms, computing such quantities can be done with a minor additional computational cost, which makes Newton's method a more suitable approach for this task. In this paper, we provide a more efficient strategy to solve S-SVDD based on Newton's method. In the proposed method, we compute the Hessian matrix and use it in the optimization process, which leads to accommodating the intrinsic curvature of the function in the iterative process.

II. Newton's method-based subspace support vector data description

Let's consider a set of data points denoted by as a matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_N], \mathbf{x}_i \in \mathbb{R}^D$, where N denoted the number of instances, and each instance $\mathbf{x}_i$ is represented in a feature space of dimensionality D. The goal is to find an optimized projection matrix $\mathbf{Q} \in \mathbb{R}^{d \times D}$ for the projection of the data from D-dimensional space to a lower d-dimensional space such that the data description for one-class classification in the lower d-dimensional space can be obtained better. The data in the lower d-dimensional space is represented as

$$\mathbf{y}_i = \mathbf{Q}\mathbf{x}_i, \ \ , i = 1, \dots, N, \tag{1}$$

To encapsulate the given training data inside a closed boundary, a hypersphere is constructed around the data. The hypersphere is characterized by center $\mathbf{a}$ and radius R. The goal is to minimize the volume of the hypersphere under the constraints that most of the training data should lie inside the hypersphere, i.e.:

$$\begin{aligned} \min \quad & F(R, \mathbf{a}) = R^2 \\ \text{s.t.} \quad & \|\mathbf{Q}\mathbf{x}_i - \mathbf{a}\|_2^2 \le R^2, \ i = 1, \dots, N, \end{aligned} \tag{2}$$

The slack variables ξ_i, $i = 1, \dots, N$ are introduced to (2) for accommodating outliers in the given training set. The optimization problem now becomes

$$\begin{aligned} \min \quad & F(R, \mathbf{a}) = R^2 + C \sum_{i=1}^{N} \xi_i \\ \text{s.t.} \quad & \|\mathbf{Q}\mathbf{x}_i - \mathbf{a}\|_2^2 \le R^2 + \xi_i, \\ & \xi_i \ge 0, \ \forall i \in \{1, \dots, N\}, \end{aligned} \tag{3}$$

Where $C > 0$ is the hyperparameter used for controlling the trade between the number of instances outside the hypersphere and the volume of the hypersphere. The constraints are incorporated into the objective function by using Lagrange multipliers.

$$\begin{aligned} L = R^2 + C \sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \gamma_i \xi_i - \sum_{i=1}^{N} \alpha_i \Big(R^2 + \xi_i - \\ \mathbf{x}_i^\intercal \mathbf{Q}^\intercal \mathbf{Q} \mathbf{x}_i + 2\mathbf{a}^\intercal \mathbf{Q} \mathbf{x}_i - \mathbf{a}^\intercal \mathbf{a} \Big), \end{aligned} \tag{4}$$

with Lagrange multipliers $\alpha_i \ge 0$ and $\gamma_i \ge 0$. By setting the partial derivative to zero, we get:

$$\frac{\partial L}{\partial R} = 0 \quad \Rightarrow \quad \sum_{i=1}^{N} \alpha_i = 1 \tag{5}$$

$$\frac{\partial L}{\partial \mathbf{a}} = 0 \quad \Rightarrow \quad \mathbf{a} = \sum_{i=1}^{N} \alpha_i \mathbf{Q} \mathbf{x}_i \tag{6}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \quad \Rightarrow \quad C - \alpha_i - \gamma_i = 0 \tag{7}$$

Substituting (5)-(7) into (4), and denoting the data in lower dimensional space according to (1) we get

$$L = \sum_{i=1}^{N} \alpha_i \mathbf{y}_i^\intercal \mathbf{y}_i - \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \mathbf{y}_i^\intercal \mathbf{y}_j \alpha_j. \tag{8}$$

Solving (8) is equivalent to solving SVDD [9] in the lower d-dimensional space. Maximizing (8) will give us α values for all training instances.

A. Regularization

Analogous to S-SVDD, after determining the optimal set of α_i, $i = 1, \dots, N$, we optimize an augmented version of the Lagrangian function. We add a regularization term Ψ, expressing the class variance in the lower d-dimensional space to (8). Hence, (8) now becomes as follows

$$L = \sum_{i=1}^{N} \alpha_i \mathbf{x}_i^\intercal \mathbf{Q}^\intercal \mathbf{Q} \mathbf{x}_i - \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \mathbf{x}_i^\intercal \mathbf{Q}^\intercal \mathbf{Q} \mathbf{x}_j \alpha_j + \beta \Psi, \tag{9}$$

where, the hyperparameter β is used to control the importance of Ψ. The regularization term is defined as follows

$$\Psi = Tr(\mathbf{Q}\mathbf{X}\lambda\lambda^\intercal \mathbf{X}^\intercal \mathbf{Q}^\intercal). \tag{10}$$

Where $Tr(\cdot)$ is the trace operator, the vector $\lambda \in \mathbb{R}^N$ is utilized to control the impact of individual samples in the regularization term. The regularization term can take three forms based on the λ vector. In the first form, denoted as $\psi 1$, all training data points contribute equally to the regularization. This is achieved by replacing all elements of the λ vector with 1. In $\psi 2$, the lambda vector is replaced by the corresponding α values of the instances. In $\psi 3$, only the alpha values of the samples corresponding to the class boundary are replaced by the corresponding α and zero for other instances. When no regularization term is used in the optimization, we refer to that case as $\psi 0$.

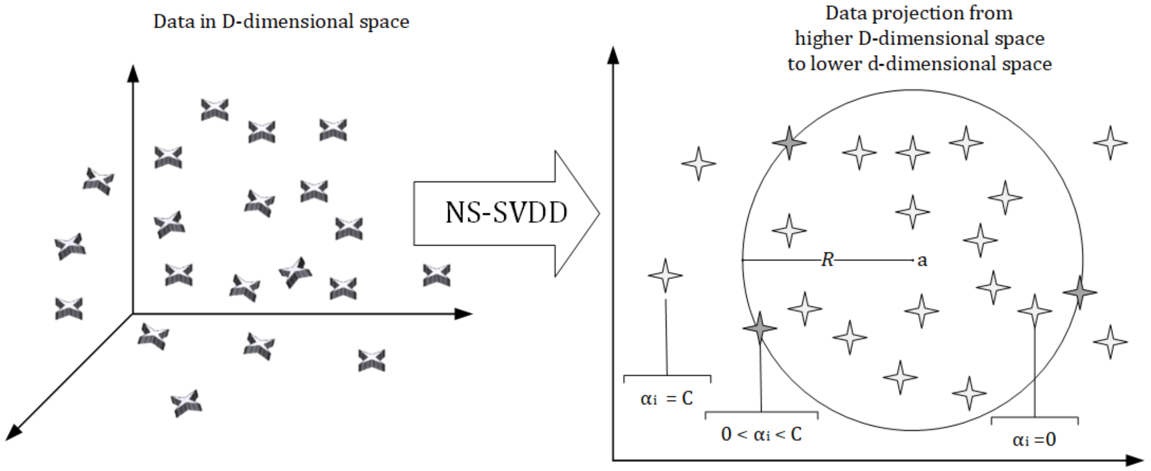
Data in D-dimensional space
NS-SVDD
Data projection from
higher D-dimensional space
to lower d-dimensional space
R
a
αi = C
0 < αi < C
αi =0

Fig. 1. An illustration of data transformation from a high-dimensional space to a lower-dimensional space, optimized for one-class classification.

B. Newton's Method-based update

We update the elements of the projection matrix $\mathbf{Q}$ iteratively using Newton's method as follows. First the Projection matrix is vectorized as $\mathcal{Q} = \text{Vec}(\mathbf{Q})$, which has the following form:

$$\mathcal{Q}^\intercal = [\mathcal{Q}_{11}, \mathcal{Q}_{12}, ..., \mathcal{Q}_{1D}, \mathcal{Q}_{21} ..., \mathcal{Q}_{dD}], \tag{11}$$

The vectorized form of the projection matrix is updated as follows:

$$\mathcal{Q} \leftarrow \mathcal{Q} - \eta[(H_L)^{-1} \Delta\mathcal{L}], \tag{12}$$

where η is the learning rate used in the optimization process and $\Delta\mathcal{L} = \text{Vec}(\Delta L)$ has the following form

$$\Delta\mathcal{L}^\intercal = [\Delta L_{11}, \Delta L_{12}, ..., \Delta L_{1D}, \Delta L_{21} ..., \Delta L_{dD}]. \tag{13}$$

The elements of (13) are computed from the gradient matrix of (9). The gradient ΔL is determined by taking the derivative of (9) with respect to the projection matrix $\mathbf{Q}$ as follows:

$$\Delta L = 2\mathbf{QXAX}^\intercal - 2\mathbf{QX}\alpha\alpha^\intercal\mathbf{X}^\intercal + \beta\Delta\Psi, \tag{14}$$

where $\Delta\Psi$ in (14) denotes the gradient of regularization term which is computed as

$$\Delta\Psi = 2\mathbf{QX}\lambda\lambda^\intercal\mathbf{X}^\intercal. \tag{15}$$

H_L in (12) is the hessian matrix,

$$H_L = \begin{vmatrix} \frac{d^2L}{d\mathcal{Q}_1\mathcal{Q}_1} & \cdots & \frac{d^2L}{\mathcal{Q}_1\mathcal{Q}_n} & \cdots & \frac{d^2L}{\mathcal{Q}_1\mathcal{Q}_{dD}} \\ \vdots & & \vdots & & \vdots \\ \frac{d^2L}{d\mathcal{Q}_m\mathcal{Q}_1} & \cdots & \frac{d^2L}{\mathcal{Q}_m\mathcal{Q}_n} & \cdots & \frac{d^2L}{\mathcal{Q}_m\mathcal{Q}_{dD}} \\ \vdots & & \vdots & & \vdots \\ \frac{d^2L}{d\mathcal{Q}_{dD}\mathcal{Q}_1} & \cdots & \frac{d^2L}{\mathcal{Q}_N\mathcal{Q}_n} & \cdots & \frac{d^2L}{\mathcal{Q}_{dD}\mathcal{Q}_{dD}} \end{vmatrix}.$$

In order to compute the elements of the H_L, we need to compute the second-order partial derivatives accordingly. We start with the first-order derivative using identities 133 and 134 in [22]:

$$\frac{\partial L}{\partial Q_{ij}} = tr\Big[\big[\frac{\partial L}{\partial \mathbf{Q}}\big]^\intercal \mathbf{S}^{ij}\Big], \tag{16}$$

where $\mathbf{S}^{ij}$ is referred to as the structure matrix with the single-entry, 1 at (i, j) and zero elsewhere. Putting (14) in (16), we get

$$\frac{\partial L}{\partial Q_{ij}} = tr(2\mathbf{XA}^\intercal\mathbf{X}^\intercal\mathbf{Q}^\intercal\mathbf{S}^{ij}) - tr(2\mathbf{X}\alpha\alpha^\intercal\mathbf{X}^\intercal\mathbf{Q}^\intercal\mathbf{S}^{ij}) \\ + tr(\beta\Delta\Psi^\intercal\mathbf{S}^{ij}) \tag{17}$$

By incorporating the gradient of regularization term, i.e., $\Delta\Psi$ into (17) and using Identity number 450 in [22], we get the following expression:

$$\frac{\partial L}{\partial Q_{ij}} = (2\mathbf{QXAX}^\intercal)_{ij} - (2\mathbf{QX}\alpha\alpha^\intercal\mathbf{X}^\intercal)_{ij} + \\ (\beta 2\mathbf{QX}\lambda\lambda^\intercal\mathbf{X}^\intercal)_{ij} \tag{18}$$

. Now, by taking the second derivative by employing identity 74 from [22], we obtain the following:

$$\frac{\partial}{\partial Q_{kl}}\Big(\frac{\partial L}{\partial Q_{ij}}\Big) = (2\mathbf{S}^{kl}\mathbf{XAX}^\intercal)_{ij} \\ -(2\mathbf{S}^{kl}\mathbf{X}\alpha\alpha^\intercal\mathbf{X}^\intercal)_{ij} + (\beta 2\mathbf{S}^{kl}\mathbf{X}\lambda\lambda^\intercal\mathbf{X}^\intercal)_{ij}. \tag{19}$$

We can further express (19) equivalently as:

$$\frac{\partial}{\partial Q_{kl}}\Big(\frac{dL}{dQ_{ij}}\Big) = 2tr\big[\mathbf{X}(\mathbf{A} - \alpha\alpha^\intercal + \lambda\lambda^\intercal)\mathbf{X}^\intercal(\mathbf{S}^{ij})^\intercal\mathbf{S}^{kl}\big] \tag{20}$$

The proof of the equality between (19) and (20) is provided in the Appendix. Utilizing (20), we can calculate the elements of the Hessian matrix H_L. The iterative process of updating the projection matrix, along with the data description, is described in the following sub-section.

C. Subspace optimization and data description

In order to optimize the subspace along with data description, an iterative two-step process is followed. In the first step, the α values are calculated using SVDD-based optimization; in the second step, the projection matrix $\mathbf{Q}$ is updated using Newton's method-based update. Recently, in [23], it is pointed out that the criterion of subspace learning for OCC can either be maximized or minimized. Hence, we provide both options for the proposed algorithms to either maximize or minimize the criterion. Algorithm 1 describes the steps involved in the subspace optimization and data description.

D. nonlinear data description

For the nonlinear data description, we utilize the nonlinear Projection Trick (NPT) [24]. In the NPT-based approach, we first compute a kernel matrix $\mathbf{K}$ using the radial basis function as

$$\mathbf{K}_{ij} = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}\right), \tag{21}$$

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

$$\hat{\mathbf{K}} = (\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^\intercal)\mathbf{K}(\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^\intercal), \tag{22}$$

where $\mathbf{1} \in \mathbb{R}^N$ is a vector with all elements set to one, and the matrix $\mathbf{I} \in \mathbb{R}^{N \times N}$ is an identity matrix. The centered kernel matrix $\hat{\mathbf{K}}$ is decomposed by using eigendecomposition:

$$\hat{\mathbf{K}} = \mathbf{U}\mathbf{A}\mathbf{U}^\intercal, \tag{23}$$

where $\mathbf{A}$ is a diagonal matrix and contains the non-negative eigenvalues of the matrix $\hat{\mathbf{K}}$ in its diagonal. The corresponding eigenvectors for the eigenvalues are stored in the columns of matrix $\mathbf{U}$. The data representation for the nonlinear data description is obtained as

$$\mathbf{\Phi} = (\mathbf{A}^{\frac{1}{2}})^{+}\mathbf{U}^{+}\hat{\mathbf{K}}, \tag{24}$$

where $+$ denotes the pseudo-inverse. After obtaining the data in Φ space, we apply all the steps used for linear data-description

E. Test phase

During testing phase, the test instantce $\mathbf{x}_*$ is first mapped to lower d-dimensional space as,

$$\mathbf{y}_* = \mathbf{Q}\mathbf{x}_*. \tag{25}$$

The decision to classify the test instance as an outlier or target class is taken on the basis of its distance from the center of the data description. The distance is calculated as follows

$$\|\mathbf{y}_* - \mathbf{a}\|_2^2 = \mathbf{y}_*^\intercal\mathbf{y}_* - 2\sum_{i=1}^{N}\alpha_i\mathbf{y}_*^\intercal\mathbf{y}_i + \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j\mathbf{y}_i^\intercal\mathbf{y}_j. \tag{26}$$

The test instance, $\mathbf{y}_*$ in the d-dimensional space is classified as positive when $\|\mathbf{y}_* - \mathbf{a}\|_2^2 \leq R^2$ and as negative, otherwise. During the test phase in the nonlinear case, the kernel vector is computed as

$$\mathbf{k}_* = \mathbf{\Phi}^\intercal\phi(\mathbf{x}_*). \tag{27}$$

The kernel vector is then centered as

$$\hat{\mathbf{k}}_* = (\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^\intercal)[\mathbf{k}_* - \frac{1}{N}\mathbf{K}\mathbf{1}]. \tag{28}$$

The centered kernel vector is then mapped to

$$\phi_* = (\mathbf{\Phi}^\intercal)^{+}\hat{\mathbf{k}}_*. \tag{29}$$

The vector ϕ_* is then treated as the test input, and the same steps as those described for the linear test are followed.

```
Input : $\mathbf{X}$, // Input data
        $\beta$ // Regularization parameter to control the
           significance of $\psi$
        $\eta$, // Learning rate parameter
        $d$, // Dimensionality of subspace
        $C$, // Regularization parameter in SVDD
        min or max // Either minimize or maximize

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

// Initialize $\mathbf{Q}$
Random initialization of $\mathbf{Q}$;
Orthogonalize $\mathbf{Q}$ using QR decomposition;
Row normalize $\mathbf{Q}$ using $l_2$ norm;
Initialize $k = 1$;

while $k < k_{max}$ do

    // SVDD in the subspace defined by $\mathbf{Q}$
    Calculate $\mathbf{Y}$ using (1);
    Calculate $\alpha_i$, $i = 1, \dots, N$ using (8);

    // Newton-based update
    Calculate $\Delta L$ using (14);
    Vectorize $\Delta L$ as $\Delta\mathcal{L} = \mathrm{Vec}(\Delta L)$ using (13);
    Vectorize $\mathbf{Q}$ as $\mathcal{Q} = \mathrm{Vec}(\mathbf{Q})$ using (11)
    Compute the elements of $H_L$ using (20);

    if minimization
      $\mathcal{Q} \leftarrow \mathcal{Q} - \eta[(H_L)^{-1}\Delta\mathcal{L}]$ ;
    elseif maximization
      $\mathcal{Q} \leftarrow \mathcal{Q} + \eta[(H_L)^{-1}\Delta\mathcal{L}]$ ;

    // Compute the de-vectorized form of the
      projection matrix $\mathbf{Q}$ and normalize
    for $i = 1 : d$ do
        for $j = 1 : D$ do
            $\mathbf{Q}[i, j]$ = [$\mathcal{Q}$[(i-1)×D + j];
        end
    end
    Orthogonalize $\mathbf{Q}$ using QR decomposition;
    Row normalize $\mathbf{Q}$ using $l_2$ norm;
    $k \leftarrow k + 1$
end

// SVDD in the optimized subspace
Calculate $\mathbf{Y}$ using (1);
Calculate $\alpha_i$, $i = 1, \dots, N$ using (8);
Compute the center $\mathbf{a}$ of data description using (6);
Identify any support vector $\mathbf{s}$ having $0 < \alpha_s < C$;
```

Algorithm 1: Newton's Method based optimization of subspace support vector data description

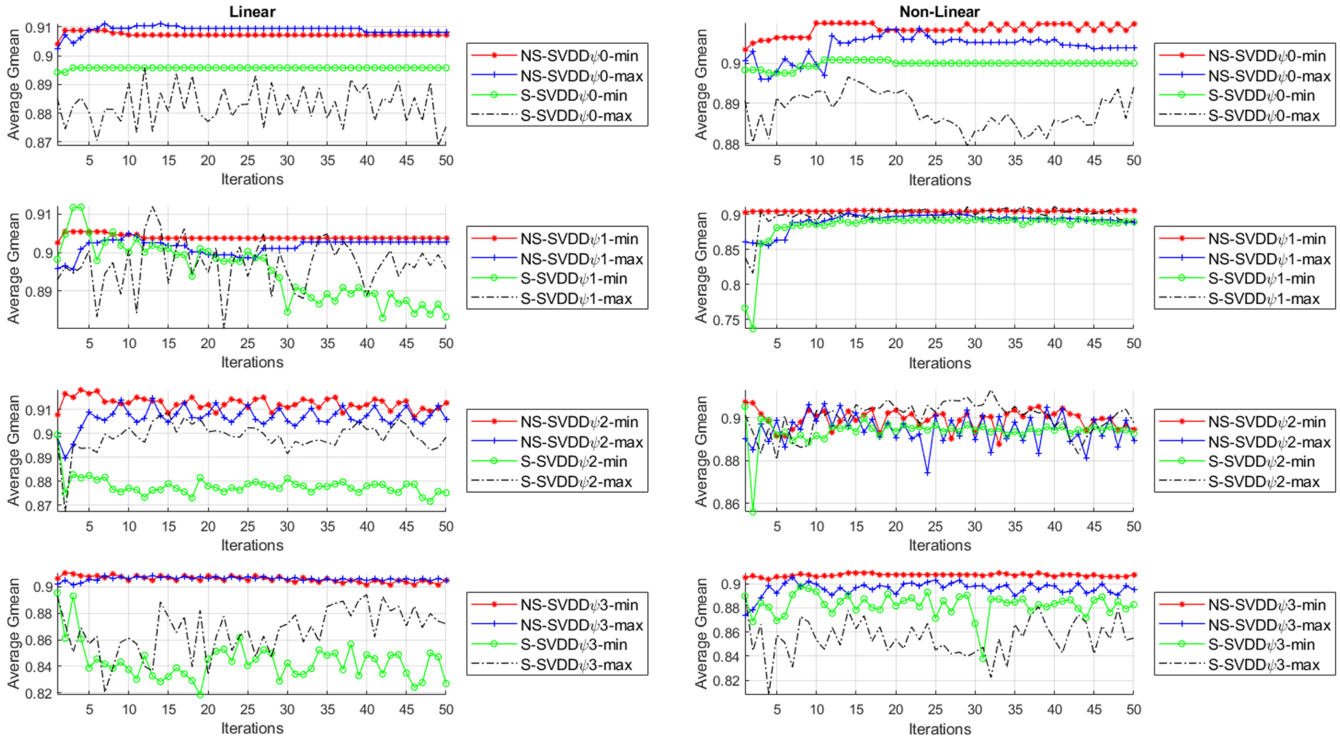

Fig. 2. Comparison of different regularization terms for Seeds dataset

III. Experiments

A. Datasets, evaluation criteria and experimental setup

For both linear and nonlinear data descriptions, we evaluated the algorithms over six different datasets downloaded from the UCI machine learning repository [25]. The datasets used in the experiments are Seeds, Qualitative bankruptcy, Somerville happiness, Iris, Ionosphere, and Sonar. The Seeds dataset consists of three classes (Kama, Rosa, Canadian), with a total of 210 samples and 70 target samples per class and a dimensionality of 7. The Qualitative bankruptcy dataset has two classes, bankruptcy (BK) and non-bankruptcy (Non-BK), with 250 total samples. The BK class has 107 target samples, the Non-BK class has 143 target samples, and the dimensionality is 6. The Somerville happiness dataset has two classes (Happy, and Un-happy) with 143 total samples. The happy class has 77 target samples, the Un-happy class has 66 target samples, and the dimensionality is also 6. The Iris dataset has three classes (Setosa, Versicolor, Virginica) with 150 samples in total and 50 target samples per class. The dimensionality for this dataset is 4. The Ionosphere dataset consists of two classes (Bad, Good) with 351 total samples. The bad class has 126 target samples, the good class has 225 target samples, and the dimensionality is 34. Finally, the Sonar dataset contains two classes (Rock, Mines) with 208 total samples. The rock class has 97 target samples, the mines class has 111 target samples, and the dimensionality is 60. We designated a single class at a time as the target class and the rest as outliers.

For each dataset, we randomly divided the data into 70% for training and the remaining 30% for testing, with the proportions of classes similar to the complete dataset. We repeated each experiment five times, using different train/test splits. We report the average test performance over the five splits. During training, we utilized 5-fold cross-validation for selecting the optimal hyperparameters based on the best Geometric mean (*Gmean*) score. *Gmean* is defined as

$$Gmean = \sqrt{tpr \times tnr}, \tag{30}$$

where tpr is the true positive rate, and tnr is the true negative rate. We chose the hyperparameters from the following range of values:

- $\beta \in \{10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$,
- $C \in \{0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$,
- $\sigma \in \{10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}\}$,
- $d \in \{1, 2, 3, 4, 5, 10, 20\}$,
- $\eta \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$.

B. Experimental results and discussion

In our evaluation, we compared the proposed NS-SVDD method with its counterpart, S-SVDD, as well as several other OCC methods. These methods include SVDD, Ellipsoidal-SVDD (E-SVDD), OC-SVM, Graph-Embedded SVM (GE-SVM), and Graph-Embedded SVDD (GE-SVDD). The performance of the proposed NS-SVDD method, along with other competing methods, is presented in Tables I and II for linear and nonlinear data description, respectively. The average performance across each dataset is reported in the ”average” (Av.) column. Additionally, for NS-SVDD

TABLE I

Gmean RESULTS FOR **LINEAR** METHODS OVER DIFFERENT DATASETS

Dataset	Seeds				Qualitative bankruptcy			Somerville happiness		
Target class	Kama	Rosa	Canadian	Av.	BK	Non-BK	Av.	Happy	Un-happy	Av.
NS-SVDDψ0-min	0.83	0.91	0.88	0.88	0.91	0.05	0.48	0.40	0.47	0.44
NS-SVDDψ1-min	0.85	0.90	0.93	0.90	0.90	0.03	0.47	0.45	0.46	0.45
NS-SVDDψ2-min	0.83	0.93	0.93	0.90	0.85	0.09	0.47	0.47	0.35	0.41
NS-SVDDψ3-min	0.82	0.91	0.88	0.87	0.90	0.21	0.55	0.46	0.41	0.43
NS-SVDDψ0-max	**0.85**	0.90	0.94	0.90	0.87	0.00	0.44	0.42	0.48	0.45
NS-SVDDψ1-max	0.82	0.91	0.93	0.88	0.90	0.00	0.45	0.38	0.46	0.42
NS-SVDDψ2-max	**0.85**	0.89	0.89	0.88	0.91	0.16	0.54	0.48	0.41	0.44
NS-SVDDψ3-max	0.81	0.90	0.77	0.83	0.87	0.19	0.53	0.39	0.43	0.41
S-SVDDψ0-min	0.85	0.92	**0.95**	**0.91**	0.91	0.21	0.56	0.47	0.41	0.44
S-SVDDψ1-min	0.84	0.93	0.94	0.90	0.82	0.24	0.53	**0.51**	0.41	0.46
S-SVDDψ2-min	0.84	0.94	0.94	**0.91**	0.93	0.16	0.55	0.49	0.43	0.46
S-SVDDψ3-min	**0.85**	0.91	0.95	0.90	0.92	0.07	0.49	0.51	0.44	0.48
S-SVDDψ0-max	**0.85**	0.93	0.94	**0.91**	0.86	0.00	0.43	0.40	0.44	0.42
S-SVDDψ1-max	**0.85**	0.92	0.94	0.90	0.86	0.00	0.43	0.42	0.45	0.44
S-SVDDψ2-max	**0.85**	**0.95**	0.94	**0.91**	0.88	0.03	0.45	0.37	0.43	0.40
S-SVDDψ3-max	0.84	0.93	0.94	0.90	0.88	0.00	0.44	0.41	0.45	0.43
E-SVDD	0.79	0.87	0.87	0.84	**0.96**	0.19	**0.58**	0.42	0.41	0.41
SVDD	**0.85**	0.92	0.94	0.90	0.94	0.00	0.47	0.41	0.36	0.39
OC-SVM	0.48	0.69	0.45	0.54	0.37	**0.41**	0.39	0.45	**0.53**	**0.49**

Dataset	Iris				Ionosphere			Sonar		
Target class	Setosa	Versicolor	Virginica	Av.	Bad	Good	Av.	Rock	Mines	Av.
NS-SVDDψ0-min	0.93	0.91	0.87	0.90	0.37	0.91	0.64	0.54	0.61	**0.58**
NS-SVDDψ1-min	0.94	0.90	0.82	0.89	0.46	**0.92**	**0.69**	0.55	0.61	**0.58**
NS-SVDDψ2-min	0.94	0.92	0.87	0.91	0.39	0.91	0.65	0.51	**0.64**	**0.58**
NS-SVDDψ3-min	0.94	0.88	0.87	0.89	0.27	**0.92**	0.60	0.51	0.58	0.55
NS-SVDDψ0-max	0.94	0.90	0.87	0.90	0.05	0.79	0.42	0.49	0.40	0.45
NS-SVDDψ1-max	0.96	0.89	0.89	0.91	0.25	0.91	0.58	**0.55**	0.44	0.50
NS-SVDDψ2-max	0.96	**0.94**	0.86	0.92	0.13	0.82	0.47	0.38	0.45	0.41
NS-SVDDψ3-max	0.90	0.91	0.89	0.90	0.18	0.87	0.52	0.40	0.41	0.40
S-SVDDψ0-min	0.94	0.92	**0.91**	**0.92**	0.16	0.79	0.48	0.50	0.52	0.51
S-SVDDψ1-min	0.94	0.92	0.88	0.91	0.16	0.81	0.49	0.48	0.54	0.51
S-SVDDψ2-min	**0.97**	0.91	0.88	0.92	0.19	0.81	0.50	0.52	0.55	0.53
S-SVDDψ3-min	0.95	0.93	0.89	0.92	0.19	0.79	0.49	0.50	0.55	0.52
S-SVDDψ0-max	0.94	0.91	0.89	0.91	0.12	0.79	0.46	0.49	0.41	0.45
S-SVDDψ1-max	0.94	0.91	0.87	0.91	0.12	0.81	0.46	0.48	0.54	0.51
S-SVDDψ2-max	0.92	0.84	0.89	0.89	0.15	0.79	0.47	0.50	0.41	0.45
S-SVDDψ3-max	0.93	0.90	0.89	0.91	0.12	0.79	0.46	0.49	0.41	0.45
E-SVDD	0.89	0.85	0.86	0.87	0.33	0.88	0.61	0.00	0.03	0.02
SVDD	0.92	0.90	0.89	0.91	0.02	0.86	0.44	0.52	0.56	0.54
OC-SVM	0.58	0.50	0.46	0.51	**0.49**	0.51	0.50	0.48	0.45	0.46

and S-SVDD, it is possible to optimize the criterion either by minimizing (-min) or maximizing (-max) it. Hence, we conducted separate experiments to maximize and minimize the overall criterion for different variants and compare the results.

The maximization strategy demonstrated that the proposed Newton-based solution outperformed the gradient-based solution in the majority of cases, both for linear and nonlinear data description. Regarding the minimization strategy, the gradient-based solution performed better in the linear case, while the Newton-based method showed superior performance in the nonlinear case.

Overall, the Newton-based method in the minimization strategy exhibited better results compared to the Newton-based method in the maximization strategy in most cases, irrespective of linear or nonlinear data description. Similarly, in the gradient-based approach, the minimization strategy yielded better outcomes in the linear case and performed equally well as the maximization strategy in the nonlinear case.

We evaluate and present the performance of the proposed NS-SVDD and the gradient-based S-SVDD methods on the test set at each training iteration for both linear and nonlinear scenarios. The average *Gmean* value is computed for each iteration across the five test splits for the Seeds dataset, which can be seen in Figure 2. We report the performance of these methods under various settings of the regularization term ψ. To compare their performance, we utilize the *Gmean* as an evaluating metric. Moreover, we report the results for both maximizing and minimizing the criterion to provide a comprehensive analysis. Similar figures can be generated for all other datasets as well. We notice that the proposed Newton-based NS-SVDD method demonstrated more stability and faster convergence to its optimal performance compared to the gradient-based S-SVDD method, as evidenced by the plotted results across different regularization terms. Considering the regularization strategies (ψ0-ψ3), the regularization strategy ψ2 was found to be effective in the linear case, while the regularization strategy ψ1 showed promising results in the nonlinear case.

TABLE II
Gmean RESULTS FOR **NONLINEAR** METHODS OVER DIFFERENT DATASETS

Dataset	Seeds				Qualitative bankruptcy			Somerville happiness		
Target class	Kama	Rosa	Canadian	Av.	BK	Non-BK	Av.	Happy	Un-happy	Av.
NS-SVDDψ0-min	0.84	0.91	0.94	0.89	0.87	0.49	0.68	0.51	0.42	0.47
NS-SVDDψ1-min	0.81	0.92	0.94	0.89	0.93	**0.62**	**0.77**	0.51	0.50	0.50
NS-SVDDψ2-min	0.83	**0.95**	0.91	0.90	0.93	0.58	0.76	0.49	0.45	0.47
NS-SVDDψ3-min	0.82	0.89	0.85	0.85	0.91	0.46	0.69	**0.57**	0.42	0.50
NS-SVDDψ0-max	**0.85**	0.92	0.94	0.90	0.94	0.50	0.72	0.43	0.46	0.44
NS-SVDDψ1-max	0.83	0.91	0.93	0.89	0.94	0.49	0.71	0.55	0.46	**0.51**
NS-SVDDψ2-max	0.83	0.91	0.94	0.89	0.94	0.49	0.71	0.42	0.31	0.37
NS-SVDDψ3-max	0.81	0.92	0.92	0.88	0.93	0.48	0.70	0.47	0.46	0.47
S-SVDDψ0-min	0.83	0.94	0.94	0.90	0.91	0.47	0.69	0.48	0.37	0.43
S-SVDDψ1-min	**0.85**	0.90	0.91	0.89	0.92	0.53	0.73	0.34	**0.52**	0.43
S-SVDDψ2-min	0.82	0.91	0.94	0.89	0.93	0.46	0.69	0.47	0.36	0.42
S-SVDDψ3-min	0.82	0.93	0.94	0.90	0.93	0.43	0.68	0.51	0.39	0.45
S-SVDDψ0-max	**0.85**	0.94	0.94	**0.91**	0.94	0.51	0.72	0.44	0.47	0.46
S-SVDDψ1-max	**0.85**	0.93	0.94	**0.91**	0.94	0.51	0.72	0.43	0.48	0.46
S-SVDDψ2-max	0.84	0.91	0.94	0.89	0.90	0.39	0.65	0.40	0.39	0.40
S-SVDDψ3-max	0.84	0.94	0.94	**0.91**	0.93	0.51	0.72	0.44	0.46	0.45
E-SVDD	0.81	0.88	0.87	0.85	0.00	0.00	0.00	0.00	0.31	0.16
SVDD	**0.85**	0.91	**0.95**	0.90	0.33	0.28	0.31	0.40	0.32	0.36
OC-SVM	0.47	0.60	0.45	0.51	0.36	0.58	0.47	0.47	0.49	0.48
GE-SVDD	0.85	0.93	0.93	0.90	0.94	0.28	0.61	0.50	0.48	0.49
GE-SVM	**0.85**	0.90	0.93	0.89	**0.95**	0.26	0.60	0.52	0.48	0.50

Dataset	Iris				Ionosphere			Sonar		
Target class	Setosa	Versicolor	Virginica	Av.	Bad	Good	Av.	Rock	Mines	Av.
NS-SVDDψ0-min	0.93	0.82	0.90	0.88	0.50	0.90	0.70	**0.59**	**0.64**	**0.62**
NS-SVDDψ1-min	0.93	**0.93**	0.85	0.90	0.63	0.90	0.77	0.57	0.56	0.56
NS-SVDDψ2-min	0.94	0.92	0.84	0.90	0.68	0.90	**0.79**	0.50	0.58	0.54
NS-SVDDψ3-min	0.89	0.83	**0.93**	0.88	0.59	0.89	0.74	0.56	0.60	0.58
NS-SVDDψ0-max	0.94	0.91	0.88	0.91	0.32	0.78	0.55	0.49	0.47	0.48
NS-SVDDψ1-max	0.96	0.90	0.81	0.89	0.67	0.91	0.79	0.53	0.59	0.56
NS-SVDDψ2-max	0.88	0.91	0.91	0.90	**0.70**	0.81	0.76	0.45	0.48	0.46
NS-SVDDψ3-max	0.94	0.88	0.58	0.80	0.57	0.80	0.68	0.49	0.45	0.47
S-SVDDψ0-min	0.94	0.90	0.90	0.91	0.51	0.90	0.70	0.53	0.51	0.52
S-SVDDψ1-min	0.96	0.88	0.90	0.91	0.46	**0.92**	0.69	0.53	0.54	0.54
S-SVDDψ2-min	0.85	0.91	0.91	0.89	0.64	0.90	0.77	0.56	0.56	0.56
S-SVDDψ3-min	0.92	0.91	0.90	0.91	0.51	0.91	0.71	0.58	0.48	0.53
S-SVDDψ0-max	**0.96**	0.91	0.89	**0.92**	0.39	0.80	0.60	0.48	0.53	0.50
S-SVDDψ1-max	**0.96**	0.91	0.87	0.91	0.41	0.80	0.60	0.49	0.58	0.54
S-SVDDψ2-max	0.94	0.92	0.88	0.91	0.42	0.82	0.62	0.47	0.48	0.48
S-SVDDψ3-max	0.96	0.89	0.90	0.92	0.41	0.82	0.61	0.48	0.53	0.50
E-SVDD	0.68	0.84	0.83	0.78	0.37	0.88	0.63	0.55	0.52	0.53
SVDD	0.92	0.92	0.88	0.90	0.21	0.85	0.53	0.53	0.59	0.56
OC-SVM	0.56	0.26	0.55	0.46	0.52	0.47	0.49	0.47	0.55	0.51
GE-SVDD	0.83	0.92	0.89	0.88	0.38	0.88	0.63	0.55	0.60	0.57
GE-SVM	0.90	0.90	0.90	0.90	0.38	0.91	0.64	0.52	0.61	0.57

IV. CONCLUSION

In this paper, we propose a novel NS-SVDD which leverages Newton's method to enhance data mapping and data description for OCC. We defined both linear and nonlinear versions for the proposed Newton-based optimization of subspace learning for OCC. We experimented with both minimization and maximization strategies and evaluated the algorithms with different regularization terms. Our findings indicate that the Newton-based method outperformed the gradient-based method in most cases for both linear and nonlinear data description in the maximization strategy. However, in the minimization strategy, the gradient-based method performed better for linear cases, while the Newton-based method performed better for nonlinear cases. Overall, the Newton-based method with the minimization strategy performed better than the maximization strategy for both linear and nonlinear data descriptions. In the future, we will investigate the usage of Newton's method in the graph-embedded based subspace learning methods for one-class classification and extend it to multi-modal OCC [26] as well.

ACKNOWLEDGEMENT

This work was supported by the NSF-Business Finland project AMALIA. Foundation for Economic Education (Grant number: 220363) funded the work of Fahad Sohrab at Haltian.

REFERENCES

[1] X. E. Pantazi, D. Moshou, and A. A. Tamouridou, "Automated leaf disease detection in different crop species through image features analysis and one class classifiers," *Computers and electronics in agriculture*, vol. 156, pp. 96–104, 2019.

[2] L. Guo, L. Zhao, Y. Wu, Y. Li, G. Xu, and Q. Yan, "Tumor detection in mr images using one-class immune feature weighted svms," *IEEE Transactions on Magnetics*, vol. 47, no. 10, pp. 3849–3852, 2011.

[3] A. Malik, A. Haque, I. A. Khan, K. S. Bharath, and S. Siddiqui, "Support vector data description (svdd) based inverter fault diagnostic method," in *2022 Second International Conference on Power, Control and Computing Technologies (ICPC2T)*. IEEE, 2022, pp. 1–6.

[4] F. Sohrab and J. Raitoharju, "Boosting rare benthic macroinvertebrates taxa identification with one-class classification," in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2020, pp. 928–933.

[5] A. Degerli, F. Sohrab, S. Kiranyaz, and M. Gabbouj, "Early myocardial infarction detection with one-class classification over multi-view echocardiography," in *2022 Computing in Cardiology*, vol. 498, pp. 1–4, 2022.

[6] S. Kilickaya, M. Ahishali, F. Sohrab, T. Ince, and M. Gabbouj, "Hyperspectral image analysis with subspace learning-based one-class classification," in *2023 Photonics & Electromagnetics Research Symposium (PIERS)*, pp. 953–959, 2023.

[7] F. Sohrab, J. Raitoharju, A. Iosifidis, and M. Gabbouj, "Ellipsoidal subspace support vector data description," *IEEE Access*, vol. 8, pp. 122 013–122 025, 2020.

[8] B. Schölkopf, R. Williamson, A. Smola, and J. Shawe-Taylor, "Sv estimation of a distribution's support," *Advances in Neural Information Processing Systems*, vol. 12, 1999.

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

[10] F. Sohrab, J. Raitoharju, M. Gabbouj, and A. Iosifidis, "Subspace support vector data description," in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 722–727.

[11] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 1999.

[12] O. Güler, *Foundations of optimization*. Springer Science & Business Media, 2010, vol. 258.

[13] P. Xu, F. Roosta, and M. W. Mahoney, "Second-order optimization for non-convex machine learning: An empirical study," in *Proceedings of the 2020 SIAM International Conference on Data Mining*. SIAM, 2020, pp. 199–207.

[14] S. Sra, S. Nowozin, and S. J. Wright, *Optimization for machine learning*. Mit Press, 2012.

[15] P. Jain, P. Kar *et al.*, "Non-convex optimization for machine learning," *Foundations and Trends® in Machine Learning*, vol. 10, no. 3-4, pp. 142–363, 2017.

[16] C. Castera, J. Bolte, C. Févotte, and E. Pauwels, "An inertial newton algorithm for deep learning," *The Journal of Machine Learning Research*, vol. 22, no. 1, pp. 5977–6007, 2021.

[17] P. Zhong and M. Fukushima, "Regularized nonsmooth newton method for multi-class support vector machines," *Optimisation Methods and Software*, vol. 22, no. 1, pp. 225–236, 2007.

[18] J. Rafati and R. F. Marica, "Quasi-newton optimization methods for deep learning applications," *Deep Learning Applications*, pp. 9–38, 2020.

[19] C.-J. Lin, R. C. Weng, and S. S. Keerthi, "Trust region newton methods for large-scale logistic regression," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 561–568.

[20] A. S. Berahas, M. Jahani, P. Richtárik, and M. Takáč, "Quasi-newton methods for machine learning: forget the past, just sample," *Optimization Methods and Software*, vol. 37, no. 5, pp. 1668–1704, 2022.

[21] J. Zhang, "Gradient descent based optimization algorithms for deep learning models training," *arXiv preprint arXiv:1903.03614*, 2019.

[22] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," nov 2012, version 20121115. [Online]. Available: https://www.math.uwaterloo.ca/ hwolkowi/matrixcookbook.pdf

[23] F. Sohrab, A. Iosifidis, M. Gabbouj, and J. Raitoharju, "Graph-embedded subspace support vector data description," *Pattern Recognition*, vol. 133, p. 108999, 2023.

[24] N. Kwak, "Nonlinear projection trick in kernel methods: An alternative to the kernel trick," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 12, pp. 2113–2119, 2013.

[25] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[26] F. Sohrab, J. Raitoharju, A. Iosifidis, and M. Gabbouj, "Multimodal subspace support vector data description," *Pattern Recognition*, vol. 110, p. 107648, 2021.

APPENDIX

Proof that (19) can be written in the form of (20). We start by taking the derivate of (17).

$$
\begin{aligned}
\frac{\partial}{\partial Q_{kl}}\Big(\frac{\partial L}{\partial Q_{ij}}\Big) &= \frac{\partial}{dQ_{kl}} tr(2\mathbf{X}\mathbf{A}^\intercal\mathbf{X}^\intercal\mathbf{Q}^\intercal\mathbf{S}^{ij}) \\
&- \frac{\partial}{dQ_{kl}} tr(2\mathbf{X}\alpha\alpha^\intercal\mathbf{X}^\intercal\mathbf{Q}^\intercal\mathbf{S}^{ij}) \\
&+ \frac{\partial}{dQ_{kl}} tr(2\beta\mathbf{X}\lambda\lambda^\intercal\mathbf{X}^\intercal\mathbf{Q}^\intercal\mathbf{S}^{ij}) \\
&= tr\Big[\big[\frac{\partial}{\partial \mathbf{Q}} tr(2\mathbf{X}\mathbf{A}^\intercal\mathbf{X}^\intercal\mathbf{Q}^\intercal\mathbf{S}^{ij})\big]^\intercal \frac{\partial \mathbf{Q}}{dQ_{kl}}\Big] \\
&- tr\Big[\big[\frac{\partial}{\partial \mathbf{Q}} tr(2\mathbf{X}\alpha\alpha^\intercal\mathbf{X}^\intercal\mathbf{Q}^\intercal\mathbf{S}^{ij})\big]^\intercal \frac{\partial \mathbf{Q}}{\partial Q_{kl}}\Big] \\
&+ tr\Big[\big[\frac{\partial}{\partial \mathbf{Q}} tr(2\beta\mathbf{X}\lambda\lambda^\intercal\mathbf{X}^\intercal\mathbf{Q}^\intercal\mathbf{S}^{ij})\big]^\intercal \frac{\partial \mathbf{Q}}{dQ_{kl}}\Big] \\
&= tr\Big[\big[2\mathbf{S}^{ij}\mathbf{X}\mathbf{A}^\intercal\mathbf{X}^\intercal\big]^\intercal\mathbf{S}^{kl}\Big] \\
&- tr\Big[\big[2\mathbf{S}^{ij}\mathbf{X}\alpha\alpha^\intercal\mathbf{X}^\intercal\big]^\intercal\mathbf{S}^{kl}\Big] \\
&+ tr\Big[\big[2\beta\mathbf{S}^{ij}\mathbf{X}\lambda\lambda^\intercal\mathbf{X}^\intercal\big]^\intercal\mathbf{S}^{kl}\Big] \\
&= 2tr\Big[\mathbf{X}\mathbf{A}\mathbf{X}^\intercal(\mathbf{S}^{ij})^\intercal\mathbf{S}^{kl}\Big] \\
&- 2tr\Big[\mathbf{X}\alpha\alpha^\intercal\mathbf{X}^\intercal(\mathbf{S}^{ij})^\intercal\mathbf{S}^{kl}\Big] \\
&+ 2\beta tr\Big[\mathbf{X}\lambda\lambda^\intercal\mathbf{X}^\intercal(\mathbf{S}^{ij})^\intercal\mathbf{S}^{kl}\Big] \\
&= 2tr\Big[\mathbf{X}\mathbf{A}\mathbf{X}^\intercal(\mathbf{S}^{ij})^\intercal\mathbf{S}^{kl} \\
&- \mathbf{X}\alpha\alpha^\intercal\mathbf{X}^\intercal(\mathbf{S}^{ij})^\intercal\mathbf{S}^{kl} \\
&+ \beta\mathbf{X}\lambda\lambda^\intercal\mathbf{X}^\intercal(\mathbf{S}^{ij})^\intercal\mathbf{S}^{kl}\Big] \\
&= 2tr\big[\mathbf{X}(\mathbf{A} - \alpha\alpha^\intercal + \lambda\lambda^\intercal)\mathbf{X}^\intercal(\mathbf{S}^{ij})^\intercal\mathbf{S}^{kl}\big]
\end{aligned}
$$