Graph Embedded One-Class Classifiers for media data classification

Vasileios Mygdalis[a], Alexandros Iosifidis[a,b], Anastasios Tefas[a], Ioannis Pitas[a,c],

[a]*Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, 54124, Greece*
[b]*Department of Signal Processing, Tampere University of Technology, FI-33101 Tampere, Finland*
[c]*Department of Electical and Electronic Engineering, University of Bristol, UK*

Abstract

This paper introduces the Graph Embedded One-Class Support Vector Machine and Graph Embedded Support Vector Data Description methods. These methods constitute novel extensions of the One-Class Support Vectors Machines and Support Vector Data Description, incorporating generic graph structures that express geometric data relationships of interest in their optimization process. Local or global relationships between the training patterns can be expressed with single graphs or combinations of fully connected and kNN graphs. We show that the adoption of generic geometric class information acts as a regularizer to the solution of the original methods. Moreover, we prove that the regularized solutions for both One-Class Support Vector Machine and Support Vector Data Description are equivalent to applying the original methods in a transformed (and shared) feature space. Qualitative and quantitative evaluation of the proposed methods shows that they compare favorably to the standard OC-SVM and SVDD classifiers, respectively.

Keywords: Media data classification, One-Class Support Vector Machine, Support Vector Data Description, Graph-based Regularization.

1. Introduction

Media data classification involves the analysis of video streams in order to extract semantic visual information related to, e.g., face recognition, human action recognition, video summarization and many other visual data classification tasks. In some cases, recognizing a specific class (usually called target class), e.g., the recognition of the leading actor in a movie, is more important that distinguishing any other classes (in such an application data forming such classes can be considered as outliers). In addition, classes of interest are usually easier to sample and annotate. In our previous example, it is expected that the leading actor will appear multiple times during the entire movie, and he (or she) will be easier to be identified by an annotator. In order to efficiently model a class of interest in media classification tasks, we consider the use of One-Class Classification (OCC) methods [1, 2, 3, 4, 5]. Related OCC applications include hyperspectral image classification [3], video summarization [6, 7], image segmentation [8]. Other OCC use case scenarios include applications when only one class is well sampled and must be distinguished from every other possibility, e.g., medical diagnostic problems, faults and failure detection, video surveillance and mobile fraud detection [9].

Perhaps the two most successful OCC methods are the One-Class Support Vector Machine (OC-SVM) [1] and the Support Vector Data Description (SVDD) methods [2]. OC-SVM discriminates the target class from the rest of the world by calculating the optimal hyperplane, with a bias term, such that all training data patterns are classified to the target class. The optimal hyperplane can be expressed as a linear combination of the training data patterns that fall close (or on) the hyperplane, i.e., the so-called support vectors. SVDD follows a similar approach, where

the target class is modeled by calculating the minimum bounding hypersphere that encloses all (or most of the) training patterns. The support vectors in this case are the training patterns that lie close to the optimal hypersphere enclosing most of the training data and satisfying a constraint minimizing its volume. Both OC-SVM and SVDD have been successfully applied in many one-class classification problems, where class uni-modality (either in the input or in the kernel space) is assumed. However, since the solution of both methods exploits information of samples belonging to the class boundaries, the obtained solutions do not consider the class distribution [10], rendering them sub-optimal for multi-modal classes and in the appearance of outliers.

In multi-class classification tasks, learning a discriminant space by considering the data distribution is important to achieve increased classification performance. To this end, the low-dimensional projection can be optimized by maximizing the geometric mean of the divergences and normalized divergence sbetween the different pairs of classes, at the same time [11]. Especially in image classification, the dimensionality of the feature space may be higher than the number of training samples, thus tensor-based image representation combined with subspace learning have been proposed in [12]. Besides the class distribution, multi-modal information can be incorporated through graph-based learning methods [13, 14]. For example, different features can be combined with labeling information under a probabilistic framework, in which the probability distributions express high-order distances between data points [13]. Data multi-modality can also be expressed in a space of reduced dimensionality which incorporates geometric pairwise information between uni-modal [15] or multi-modal descriptors [16]. Uni-modal or multi-modal information contained in graph structures have been also exploited

in recently proposed multi-class classification methods [17, 18], with decent results. Moreover, information contained in graph structures has been exploited in the semi-supervised classification case. In [19], kNN graphs are employed for manifold regularization, so that local geometric data relationships between labeled and unlabeled data are expressed in the deformed space. The corresponding semi-supervised OCC method has been proposed in [3].

In the OCC case, the exploitation of within-class multi-modal information in order to improve classification performance, has not been thoroughly considered. However, manifold regularization techniques have been proposed to this end. Considering the data distribution in the optimization process of the OC-SVM [10] and SVDD [20], e.g. by employing the data covariance matrix, leads to a regularized solution that emphasizes on the low variance directions. However, there might be cases where the target class will form multiple subclasses which are related to, e.g., illumination changes or different viewing angles [18, 21]. Thus, employing the methods proposed in [10] and [20] does not model the subclass properties of the target class, sufficiently. To this end, one could employ kNN graphs in the OC-SVM optimization process as in [3] (for the supervised classification case). By employing kNN graphs, local geometric data relationships between the training data may be sufficiently modeled. However, since there is no consideration about the data distribution in the optimization process, the obtained solution might be sub-optimal.

In this paper, we describe a generic Graph Embedding framework for OC-SVM and SVDD, that models intrinsic geometric data information of the target class in the OC-SVM and SVDD optimization process. In order to exploit such geometric information in the proposed framework, we employ graph structures

that describe local or global relationships between the training patterns, or combinations of fully connected and kNN graphs. Using this framework, the standard OC-SVM and SVDD methods [1, 2], as well as the covariance based OC-SVM and SVDD approaches [10, 20] can be considered as special cases of the proposed approach, when a specific type of fully connected graph is employed. Moreover, the proposed method allows us to create graphs which describe subclass information, e.g. by applying data clustering. In addition, by employing a generic description of the data relationships being exploited in both optimization problems, the proposed methods can be directly applied by using any kind of graph structure, allowing us to exploit a priori information for the problem to be solved. Such graphs can either be automatically created (e.g. the k-NN graph) or be designed specifically for the targeted application (e.g. a hand-crafted graph provided by a human expert).

We show that the adoption of generic geometric class information in one-class classification has the effect of regularization. Moreover, we prove that the obtained regularized solutions for both One-Class Support Vector Machine and Support Vector Data Description are equivalent to mapping the input space to a new feature space of specific structure (which is the same for both OC-SVM and SVDD). At that feature space, the application of the original methods is equivalent to the application of the regularized methods in the input space. This analysis verifies the findings of prior works [22, 23] denoting that the two methods are closely related. In addition, it allows us to use efficient OC-SVM and SVDD implementations for obtaining general solutions incorporating geometric data information in one-class classification problems [24, 25].

In summary, the contributions of the paper are as follows:

- Two novel extensions of the OC-SVM and SVDD methods which can exploit generic data relationships encoded in graph structures are proposed.
- We show that the solutions of the proposed GE-OC-SVM and GE-SVDD methods is equivalent to the solution of the original methods in a transformed feature space. Moreover, we prove that this transformed kernel space is the same for both GE-OC-SVM and GE-SVDD methods.
- We show that geometric data relationships encoded in multiple graph structures can also be employed in order to regularize the solution of the GE-OC-SVM and GE-SVDD methods.
- We evaluate the proposed methods and compare their performance with related ones in a wide range of applications, i.e. face recognition, human action recognition, video summarization and generic one-class classification problems. The proposed methods compare favourably to the competing ones.

The remainder of this paper is structured as follows. In Section 2, we describe in detail the proposed methods and provide a discussion explaining their connection with other methods. Experiments conducted in order to evaluate the performance of the proposed approach are provided in Section 3. Finally, conclusions are drawn in Section 4.

2. Method Description

In this section, we start by briefly describing the Graph Embedding framework for supervised subspace learning in Subsection 2.1. Subsequently, in Subsections

2.2 and 2.3, we describe in detail the proposed Graph Embedding One-Class SVM and Graph Embedding SVDD methods, respectively. Next, in Subsection 2.4, we describe how multiple Graph types can be combined in the GE-OC-SVM and GE-SVDD optimization processes. Finally, in Subsection 2.5, we discuss the connection of the proposed methods with other related methods, the differences between them, as well as the computational complexity of the proposed methods. Important notations used in the entire paper are summarized in Table 1.

Table 1: Nomenclature

Notations	Descriptions
$\mathbf{x}_i \in \mathbb{R}^D, i = 1, \ldots, N$	A dataset consisting of N training patterns $\mathbf{x}_i$, with feature dimensionality D.
$\phi(\cdot) : \mathbb{R}^D \mapsto \mathcal{F}$	Any non linear function that maps the training patterns from the input space to the kernel space.
$\mathbf{\Phi} \in \mathcal{F}, \mathbf{\Phi} = [\phi(\mathbf{x}_i, \ldots, \phi(\mathbf{x}_N)]$	A matrix that contains the training data representations in $\mathcal{F}$.
$\mathbf{K} \in \mathbb{R}^{N \times N}, \mathbf{K} = \mathbf{\Phi}^T\mathbf{\Phi}$	The kernel matrix which contains dot products between the training pattern representations in $\mathcal{F}$.
$\mathbf{w}$	The OC-SVM hyperplane.
$\mathbf{a}$, R	The hypershere center $\mathbf{a}$ and Radius R of SVDD.
λ_i, ℓ_i	The Lagrange multipliers corresponding to the constraints of the OC-SVM and SVDD optimization problems.
$\boldsymbol{\beta} \in \mathbb{R}^N$	A reconstruction vector employed to represent the hyperplane $\mathbf{w}$ in $\mathcal{F}$, such that $\mathbf{w} = \mathbf{\Phi}\boldsymbol{\beta}$.
$\mathcal{G} = \{\mathbf{\Phi}, \mathbf{A}\}$	Undirected weighted graph, describing connections in $\mathcal{F}$ between the training data representations $\mathbf{\Phi}$.
$\mathbf{A}, \mathbf{D}, \mathbf{L} \in \mathbb{R}^{N \times N}$	$\mathbf{A}$ is the graph weight matrix, $\mathbf{D}$ is the Degree and $\mathbf{L}$ is the Laplacian matrix.
$\mathbf{S} = \mathbf{\Phi}\mathbf{L}\mathbf{\Phi}^T$	Matrix encoding geometric data relationships in the GE-OC-SVM and GE-SVDD optimization processes.

2.1. Graph Embedding

In this section, we describe how geometric data relationships can be expressed by employing the graph embedding framework [15], which has been proposed for dimensionality reduction and manifold learning. This approach can also be used to describe geometric data relationships for the supervised learning case [17, 18, 26]. Let $\mathcal{G} = \{\mathbf{\Phi}, \mathbf{A}\}$ be an undirected weighted graph, where it is assumed that the training data representations in $\mathcal{F}$, i.e. $\mathbf{\Phi} = [\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_N)]$, form the vertex set of the graph and $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the graph weight matrix. The graph Laplacian matrix $\mathbf{L} \in \mathbb{R}^{N \times N}$ is defined by $\mathbf{L = D - A}$, where $\mathbf{D} \in \mathbb{R}^{N \times N}$ is the (diagonal)

degree matrix having elements $[\mathbf{D}]_{ii} = \sum_{i \neq j} [\mathbf{A}]_{ij}$, $i = 1, \ldots, N$. $\mathbf{L}$ can be used in order to describe geometric data relationships employed in several dimensionality reduction and manifold learning techniques, such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Clustering-based Discriminant Analysis (CDA), Laplacian Eigenmap (LE) and Locally Linear Embedding (LLE) [15]. For example, the scatter of the training data used in PCA can be expressed by:

$$\mathbf{S}_T = \frac{1}{N} \mathbf{\Phi} \left(\mathbf{I} - \frac{1}{N} \mathbf{1}\mathbf{1}^T \right) \mathbf{\Phi}^T = \mathbf{\Phi} \mathbf{L}_T \mathbf{\Phi}^T, \tag{1}$$

where $\mathbf{I} \in \mathbb{R}^{N \times N}$ is the identity matrix and $\mathbf{1} \in \mathbb{R}^N$ is a vector of ones.

In the case where the data form subclasses, e.g. in visual-based activity recognition applications when the data correspond to different activities or camera view angles, subclasses can be determined in the feature space by employing the kernel K-means algorithm [27]. In this case the scatter of the training data, forming Z groups or subclasses, can be expressed by:

$$\mathbf{S}_w = \mathbf{\Phi} \left(\mathbf{I} - \sum_{z=1}^{Z} \frac{1}{N_z} \mathbf{1}_z \mathbf{1}_z^T \right) \mathbf{\Phi}^T = \mathbf{\Phi} \mathbf{L}_w \mathbf{\Phi}^T, \tag{2}$$

where N_z is the total number of patterns belonging to cluster z and $\mathbf{1}_z \in \mathbb{R}^N$ is a vector having ones in its elements that correspond to data belonging to subclass z and zeros everywhere else.

In order to exploit local geometric information, pair-wise similarities between the graph vertices can be expressed by adopting the heat kernel function:

$$a_{ij} = exp \left(- \frac{||\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)||_2^2}{2\sigma^2} \right), \tag{3}$$

where σ is a parameter scaling the Euclidean distance between $\phi(\mathbf{x}_i)$ and $\phi(\mathbf{x}_j)$.

Finally, a kNN graph weight matrix is formed such that:

$$[\mathbf{A}]_{ij} = \begin{cases} a_{ij}, & \text{if } \phi(\mathbf{x}_j) \in \mathcal{N}_i \\ 0, & \text{otherwise,} \end{cases}$$

where $\mathcal{N}_i$ denotes the neighborhood of $\mathbf{x}_i$ in the feature space $\mathcal{F}$. Subsequently, the scatter matrix describing local data relationships is obtained by:

$$\mathbf{S}_{kNN} = \mathbf{\Phi}(\mathbf{D} - \mathbf{A})\mathbf{\Phi}^T = \mathbf{\Phi}\mathbf{L}_{kNN}\mathbf{\Phi}^T. \tag{4}$$

By comparing (1), (2) and (4), we can observe that both global and local geometric data relationships can be expressed by using a generic matrix of the form:

$$\mathbf{S}_X = \mathbf{\Phi}\mathbf{L}_X\mathbf{\Phi}^T, \tag{5}$$

where the subscript X denotes the adopted graph type. In what follows, we drop the subscript X for notation simplicity. We will employ such a matrix in order to formulate the proposed Graph Embedded OC-SVM and Graph Embedded SVDD classifiers in the following Subsections.

2.2. Graph Embedded One-Class Support Vector Machines

Within the one-class classification framework, we would like to obtain a decision hyperplane $\mathbf{w} \in \mathcal{F}$, by using an optimization problem that exploits geometric data relationships expressed in a matrix $\mathbf{S}$ (5). We propose the Graph Embedded One-Class Support Vector Machines (GE-OC-SVM) optimization problem to this end:

$$\min_{\mathbf{w},\xi_i,\rho} \frac{1}{2}\mathbf{w}^T\mathbf{S}\mathbf{w} + \frac{1}{\nu N}\sum_{i=1}^{N}\xi_i - \rho \tag{6}$$

$$s.t. : \mathbf{w}^T\phi(\mathbf{x}_i) \geq \rho - \xi_i, \;\; i = 1, \ldots, N, \tag{7}$$

$$\xi_i \geq 0, \;\; i = 1, \ldots, N, \tag{8}$$

$\xi_i,\ i = 1, \ldots, N$ are the slack variables, ρ is the bias term and $\nu > 0$ is a parameter that defines a trade-off between minimizing the two terms. An additional constraint $\mathbf{w}^T\mathbf{S}\mathbf{w} \geq 0$ is also imposed to ensure the positive definiteness of $\mathbf{S}$. The dimensionality of $\mathcal{F}$ is determined by the adopted kernel function choice. For example, the dimensionality of $\mathcal{F}$ in the case where the linear kernel function is used is D, while it is infinite when the RBF kernel function is used. In the case where $\mathcal{F}$ is of arbitrary dimensions, $\mathbf{S}$ might be singular. Thus, in order to improve numerical stability, we employ a regularized version of $\mathbf{S}$ such that:

$$\tilde{\mathbf{S}} = \mathbf{\Phi}\mathbf{L}\mathbf{\Phi}^T + r\mathbf{I}, \tag{9}$$

where r is a regularization parameter ensuring the positive definiteness of $\mathbf{S}$ and $\mathbf{I}$ is the identity matrix of appropriate dimensions. Based on Representer theorem [28], the non-linear decision hyperplane $\mathbf{w}$ can be expressed as a linear combination of the training data representations in $\mathcal{F}$, by using a reconstruction vector $\boldsymbol{\beta} \in \mathbb{R}^N$ such that:

$$\mathbf{w} = \mathbf{\Phi}\boldsymbol{\beta}. \tag{10}$$

Thus, the optimization problem in (6)-(8) can be redefined by using (9) and (10) as follows:

$$\min_{\boldsymbol{\beta},\xi_i,\rho} \frac{1}{2}\boldsymbol{\beta}^T(\mathbf{K}\mathbf{L}\mathbf{K} + r\mathbf{K})\boldsymbol{\beta} + \frac{1}{\nu N}\sum_{i=1}^{N}\xi_i - \rho \tag{11}$$

$$s.t. : \boldsymbol{\beta}^T\mathbf{k}_i \geq \rho - \xi_i,\ \ i = 1, \ldots, N, \tag{12}$$

$$\xi_i \geq 0, \tag{13}$$

where $\mathbf{K} = \mathbf{\Phi}^T\mathbf{\Phi}$ is the kernel matrix and $\mathbf{k}_i$ is a vector containing the dot products between the i-th training pattern with all the training patterns in $\mathcal{F}$ (i.e. the

i-th column of $\mathbf{K}$). Based on KKT conditions, this optimization problem can be solved by determining the saddle points of the Lagrangian:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\beta}, \xi_i, \rho) =& \frac{1}{2}\boldsymbol{\beta}^T(\mathbf{KLK} + r\mathbf{K})\boldsymbol{\beta} + \frac{1}{\nu N}\sum_{i=1}^{N}\xi_i - \rho \\ & - \sum_{i=1}^{N}\lambda_i\left(\boldsymbol{\beta}^T\mathbf{k}_i - \rho + \xi_i\right) - \sum_{i=1}^{N}\ell_i\xi_i, \end{aligned} \tag{14}$$

where λ_i and ℓ_i are the Lagrange multipliers corresponding to the constraints (27) and (28). The optimization can be achieved when the following optimality conditions are met:

$$\frac{\vartheta\mathcal{L}(\boldsymbol{\beta}, \xi_i, \rho)}{\vartheta\boldsymbol{\beta}} = 0 \Rightarrow (\mathbf{KLK} + r\mathbf{K})\boldsymbol{\beta} = \sum_{i=1}^{N}\lambda_i\mathbf{k}_i, \tag{15}$$

$$\frac{\vartheta\mathcal{L}(\boldsymbol{\beta}, \xi_i, \rho)}{\vartheta\xi_i} = 0 \Rightarrow \ell_i = \frac{1}{\nu N} - \lambda_i, \tag{16}$$

$$\frac{\vartheta\mathcal{L}(\boldsymbol{\beta}, \xi_i, \rho)}{\vartheta\rho} = 0 \Rightarrow \sum_{i=1}^{N}\lambda_i = 1. \tag{17}$$

From (15), the reconstruction vector $\boldsymbol{\beta}$ is given by:

$$\boldsymbol{\beta} = (\mathbf{KLK} + r\mathbf{K})^{-1}\mathbf{K}\boldsymbol{\lambda}, \tag{18}$$

where $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_N]^T$ is a vector containing the Lagrange multipliers. Thus, the optimal hyperplane $\mathbf{w}$ can be calculated through (10). The training vectors $\mathbf{x}_i$ corresponding to $\lambda_i \neq 0$ are called support vectors. In order to recover the term ρ we can employ any support vector $\mathbf{x}_i$ whose co-efficient λ_i satisfies $0 < \lambda_i < \frac{1}{\nu N}$:

$$\rho = \mathbf{w}^T\phi(\mathbf{x}_i) = \sum_{j=1}^{N}\beta_j\kappa(\mathbf{x}_j, \mathbf{x}_i), \tag{19}$$

where $\kappa(\mathbf{x}_j, \mathbf{x}_i)$ expresses data similarity in $\mathcal{F}$ between $\mathbf{x}_j$ and $\mathbf{x}_i$. By substituting (15), (16) and (17) in (14) the Lagrangian function of GE-OC-SVM problem takes

the following form:

$$\mathcal{L} = -\frac{1}{2}\boldsymbol{\lambda}^T\mathbf{K}(\mathbf{KLK} + r\mathbf{K})^{-1}\mathbf{K}\boldsymbol{\lambda}. \tag{20}$$

Finally, the response of the classifier for a test pattern $\mathbf{v}_t \in \mathbb{R}^D$ is given by:

$$f(\mathbf{v}_t) = \mathbf{w}^T\phi(\mathbf{v}_t) - \rho = \boldsymbol{\beta}^T\mathbf{k_t} - \rho, \tag{21}$$

where $\mathbf{k}_t = \boldsymbol{\Phi}^T\phi(\mathbf{v}_t) = [\kappa(\mathbf{v}_t, \mathbf{x}_1), \ldots, \kappa(\mathbf{v}_t, \mathbf{x}_N)]^T$ is a vector that contains the dot products of the test pattern $\mathbf{v}_t$ with all training patterns $\mathbf{x}_i$ in $\mathcal{F}$. The test pattern $\mathbf{v}_t$ is classified to the target class when $f(\mathbf{v_t}) \geq 0$.

In order to employ standard OC-SVM implementations, we can employ a deformed version of the kernel $\mathbf{K}$, such that:

$$\tilde{\kappa}(\mathbf{x}_j, \mathbf{x}_i) = \frac{1}{r}\left[\mathbf{I} - (\mathbf{L} + r\mathbf{K}^{-1})^{-1}\mathbf{L}\right]\kappa(\mathbf{x}_j, \mathbf{x}_i). \tag{22}$$

All auxiliary steps followed to derive (22) can be found in Appendix A.

2.3. Graph Embedded SVDD

As in GE-OC-SVM, we would like to exploit geometric data relationships within the SVDD optimization problem. We follow the same approach as before by exploiting a generic graph structure defining the (regularized) matrix $\tilde{\mathbf{S}}$ that expresses geometric information as in (9). We would like to calculate the minimum bounding hypershere that encloses (most of) the training data representations in $\mathcal{F}$, where $\mathbf{a}$ is the hypershere center and R the minimum radius. We formulate the proposed Graph Embedded SVDD (GE-SVDD) optimization problem as follows:

$$\min_{R,\xi_i,\mathbf{u}} R^2 + c\sum_{i=1}^{N}\xi_i \tag{23}$$

$$s.t.: \Big(\phi(\mathbf{x}_i) - \mathbf{a}\Big)^T\tilde{\mathbf{S}}^{-1}\Big(\phi(\mathbf{x}_i) - \mathbf{a}\Big) \leq R^2 + \xi_i, \tag{24}$$

$$\xi_i \geq 0, \qquad i = 1, \ldots, N, \tag{25}$$

where ξ_i, $i = 1, \ldots, N$ are the slack variables and $c > 0$ is a parameter that affects the optimal radius R by allowing some training errors. As can be observed from (24), the proposed GE-SVDD problem defines a distance function in which each dimension is appropriately scaled based on the matrix $\tilde{\mathbf{S}}$ expressing geometric data relationships of interest. By defining a vector $\mathbf{u} = \tilde{\mathbf{S}}^{\frac{1}{2}}\mathbf{a}$, the problem in (23)-(25) can be expressed as follows:

$$\min_{R,\xi_i,\mathbf{u}} R^2 + c\sum_{i=1}^{N}\xi_i \tag{26}$$

$$s.t. : \|\tilde{\mathbf{S}}^{-\frac{1}{2}}\phi(\mathbf{x}_i) - \mathbf{u}\|_2^2 \leq R^2 + \xi_i, \tag{27}$$

$$\xi_i \geq 0, \qquad i = 1, \ldots, N, \tag{28}$$

Based on (KKT) conditions, the corresponding Lagrangian function is given by:

$$\mathcal{L}(R, \xi, \mathbf{u}) = R^2 + c\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\ell_i\xi_i -$$
$$-\sum_{i=1}^{N}\lambda_i\left(R^2 + \xi_i - \|\tilde{\mathbf{S}}^{-\frac{1}{2}}\phi(\mathbf{x}_i) - \mathbf{u}\|_2^2\right), \tag{29}$$

where λ_i and ℓ_i are the Lagrange multipliers corresponding to the constraints (27) and (28).

By calculating the saddle points of the Lagrangian, we obtain the following optimality conditions:

$$\frac{\vartheta\mathcal{L}(R, \xi, \mathbf{u})}{\vartheta\mathbf{u}} = 0 \Rightarrow \mathbf{u} = \sum_{i=1}^{N}\lambda_i\tilde{\mathbf{S}}^{-\frac{1}{2}}\phi(\mathbf{x}_i) \tag{30}$$

$$\frac{\vartheta\mathcal{L}(R, \xi, \mathbf{u})}{\vartheta\xi_i} = 0 \Rightarrow \ell_i = c - \lambda_i, \tag{31}$$

$$\frac{\vartheta\mathcal{L}(R, \xi, \mathbf{u})}{\vartheta R} = 0 \Rightarrow \sum_{i=1}^{N}\lambda_i = 1. \tag{32}$$

From (31) and (32), (31) is satisfied when $0 \leq \lambda_i \leq c$ [2]. The vector $\mathbf{u}$ is calculated using (30), while the relative to $\mathbf{u}$ hypersphere center $\mathbf{a}$ can be calculated as follows:

$$\mathbf{a} = \tilde{\mathbf{S}}^{-1}\mathbf{\Phi}\boldsymbol{\lambda}, \tag{33}$$

where $\boldsymbol{\lambda} = [\lambda_i, \ldots, \lambda_N]^T$ is a vector containing the Lagrange multipliers and $\mathbf{\Phi} = [\phi(\mathbf{x}_i), \ldots, \phi(\mathbf{x}_N)]$ is a matrix that contains the data representations in the feature space $\mathcal{F}$.

The Lagrangian function of GE-SVDD, after replacing equations (30), (31) and (32) in (29), takes the following form:

$$\mathcal{L} = \sum_{i=1}^{N} \lambda_i \phi(\mathbf{x}_i)^T \tilde{\mathbf{S}}^{-1} \phi(\mathbf{x}_i) - \sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i \lambda_j \phi(\mathbf{x}_i)^T \tilde{\mathbf{S}}^{-1} \phi(\mathbf{x}_j). \tag{34}$$

The optimal radius R can be calculated as the distance of the hypersphere center to a support vector [2], i.e.:

$$R^2 = \{min \, \|\tilde{\mathbf{S}}^{-\frac{1}{2}} \phi\left(\mathbf{x}_i\right) - \mathbf{u}\|_2^2, \;\; \mathbf{x}_i \text{ is a SV}\}. \tag{35}$$

The response of the classifier for a test pattern $\mathbf{v}_t$ is given by:

$$f(\mathbf{v}_t) = R - \|\tilde{\mathbf{S}}^{-\frac{1}{2}} \phi(\mathbf{v}_t) - \mathbf{u}\|_2, \tag{36}$$

and $\mathbf{v}_t$ is classified to the modeled class if it falls inside the hypersphere defined by the radius R and center $\mathbf{a}$.

By observing the dual optimization problem of GE-SVDD in (34) and comparing it with that of the standard SVDD [2], we can conclude that the solution of the proposed GE-SVDD classifier is equivalent to the solution of SVDD in a transformed feature space, defined as follows:

$$\tilde{\kappa}(\mathbf{v}_t, \mathbf{v}_t) - 2\sum_{i=1}^{N} \lambda_i \tilde{\kappa}(\mathbf{v}_t, \mathbf{x}_i) + \sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i \lambda_j \tilde{\kappa}(\mathbf{x}_i, \mathbf{x}_j) \leqslant R^2, \tag{37}$$

where $\tilde{\kappa}$ is the deformed kernel space where the geometric data relationships have been expressed. As proven in Appendix B, this space is the same as the one obtained for the GE-OC-SVM classifier (22).

2.4. Graph Embedded OCC exploiting multiple graphs

The proposed GE-OC-SVM and GE-SVDD methods can exploit single or multiple graph types, in their optimization processes. In this Subsection, we demonstrate the general case of M graphs. Let M undirected graphs $G^j = \{\mathbf{\Phi}, \mathbf{A}^j\}, j = 1, \ldots, M$, expressing different reltionships between the vertex set, formed by the training data representations in the feature space $\mathbf{\Phi} = [\phi(\mathbf{x}_i), \ldots, \phi(\mathbf{x}_N)]$ and $\mathbf{A}^j$ is the j-th graph weight matrix. By employing the graph embedding framework described in Subsection 2.1, a combination of geometric data relationships described in the M graphs can be expressed in a matrix $\mathbf{S}$ as follows:

$$\begin{aligned}\mathbf{S} &= \mu_1\mathbf{S}_1 + \cdots + \mu_M\mathbf{S}_M = \\ &= \mu_1\mathbf{\Phi}\mathbf{L}_1\mathbf{\Phi}^T + \cdots + \mu_M\mathbf{\Phi}\mathbf{L}_M\mathbf{\Phi}^T \\ &= \mathbf{\Phi}\left(\mu_1\mathbf{L}_1 + \cdots + \mu_M\mathbf{L}_M\right)\mathbf{\Phi}^T, \end{aligned} \tag{38}$$

where $\mathbf{S}_j$ is the matrix formed for the $j-$th graph and $\mathbf{L}_j$ is the corresponding Laplacian matrix. The geometric data relationships of the M graphs can be combined using a weight parameter μ_j for each Laplacian matrix. In order to alleviate the effect of over-regularization caused by employing multiple graphs, we restrict the parameter $\mu_j \in [0, 1]$ and demand that $\sum_{j=1}^{M} \mu_j = 1$.

Essentially, the parameters μ_j denote the membership of each graph in the regularization. If we assume that all graph types present equally important information about the training data, μ_j can be set equal to $\mu_j = 1/M$. In the case where two graphs are employed, and we would like to set different weights in each

graph (i.e. in the case where one of the employed graph types is hand-crafted), only one parameter needs to be tuned if we set $\mu_1 = \alpha$ and $\mu_2 = 1 - \alpha$, where $\alpha \in [0, 1]$. In our experiments, we employ a combination of two automatically generated graphs, with weights α and $1 - \alpha$, respectively. First, we employ a graph that represents the class distribution with respect to subclass information, mentioned in equation (2), along with the kNN graph described in equation (4), which implies that local geometric data relationships should be preserved as well.

Finally, since $\mathbf{S}$ can be defined in feature spaces of high (or even infinite) dimensionality, we adopt a regularized version such that:

$$\tilde{\mathbf{S}} = \mathbf{\Phi} \left(\mu_1 \mathbf{L}_1 + \cdots + \mu_M \mathbf{L}_M \right) \mathbf{\Phi}^T + r\mathbf{I}, \tag{39}$$

where r is a regularization parameter. In order to employ multiple graphs in the OC-SVM and SVDD optimization processes, we define a matrix $\tilde{\mathbf{L}} \in \mathbb{R}^{N \times N}$ as follows:

$$\tilde{\mathbf{L}} = \mu_1 \mathbf{L}_1 + \cdots + \mu_M \mathbf{L}_M, \tag{40}$$

and replace it with $\mathbf{L}$ in (22) as follows:

$$\tilde{\kappa}(\mathbf{x}_t, \mathbf{x}_i) = \frac{1}{r} \left[\mathbf{I} - (\tilde{\mathbf{L}} + r\mathbf{K}^{-1})^{-1} \tilde{\mathbf{L}} \right] \kappa(\mathbf{x}_t, \mathbf{x}_i). \tag{41}$$

Finally, in order to solve the GE-OC-SVM or GE-SVDD optimization problems with multiple graphs, we can employ standard OC-SVM or SVDD implementations, by employing the deformed kernel matrix defined in (41).

2.5. Discussion

The proposed GE-OC-SVM and GE-SVDD optimization processes are equivalent to a two-step process where both training and test patterns are mapped from

the input space to a deformed kernel space followed by the application of the original OC-SVM and SVDD methods. As detailed in Appendix A and Appendix B, the derived space is the same for both GE-OC-SVM and GE-SVDD, which is inline with related literature [1, 23] showing that (in the case where a normalized kernel function is used) the solutions of OC-SVM and SVDD are identical.

In the following, we also show that the proposed approach can be seen as a general way of incorporating geometric data information in one-class classification models. This means that most of the existing methods following the OC-SVM and SVDD formulations are special cases of the proposed methods. We can identify the following cases:

- No geometric data information is used: This case corresponds to the choice of $\mathbf{L} = \mathbf{0}$, where $\mathbf{0}$ is a matrix of zeros. In this case the proposed GE-OC-SVM and GE-SVDD methods degenerate to the OC-SVM [1] and SVDD [2] methods.

- The total variance of the (unimodal) class is used: This case corresponds to the choice of $\mathbf{L} = \mathbf{L}_T$ (given in (1)). In this case the proposed GE-OC-SVM degenerates to the method of [10] and the GE-SVDD degenerates to the method of [20].

Here we should also note that local geometric information expressed by kNN graphs has also been exploited in a semi-supervised one-class classification setting, as in [3] where the Laplacian One-Class SVM (LAP-OC-SVM) is proposed. However, LAP-OC-SVM assumes that the training set is formed by data belonging to both positive and negative classes (in fact LAP-OC-SVM is modeled as a two-class classification model). The assumption of LAP-OC-SVM is that, while

samples from both classes are available during the training phase, a part of the samples forming the positive class is labeled, while the remaining data forming the positive class (and the data forming the negative class) are unlabeled. This problem is usually defined as the single-class classification problem [29]. On the contrary, the proposed GE-OC-SVM and GE-SVDD methods assume that the training set is formed only by positive data, i.e. addresses the standard one-class classification problem. By using the graph Laplacian matrix $\mathbf{L}_{kNN}$ (described in (4)), they provide a natural way of incorporating local geometric data information in one-class classification models.

Since the proposed methods can be solved using existing OC-SVM and SVDD implementations, their additional computational complexity depends on the adopted graph type. Let us consider the GE-OC-SVM case. For a dataset formed by samples $\mathbf{x}_i \in \mathbb{R}^D, i = 1, \ldots, N$, the additional steps followed to derive the GE-OC-SVM solution require:

- Deriving the Laplacian matrix $\mathbf{L}$, whose complexity depends on the adopted graph type. For example, in the case where $\mathbf{L} = \mathbf{L}_T$, only multiplications with vectors of ones are required. In the case where $\mathbf{L} = \mathbf{L}_w$, a data clustering step is required to derive the subclasses. When a heat kernel is employed in order to determine the graphs weights $\mathbf{A}$ an additional computational cost of $O(DN^2)$ is required. However, in the latter case, one can substitute the weight matrix $\mathbf{A}$ with the kernel matrix $\mathbf{K}$ and then, truncate the elements not corresponding to neighbors. This approach allows us also to address scaling issues that may appear in cases where different distance metrics are employed for $\mathbf{K}$ and $\mathbf{A}$ calculation.

- Kernel space deformation using (22). This includes a matrix multiplication

and an inversion, which are of $O(2N^3)$.

The computational complexity of GE-SVDD can be derived in a similar manner. Optimization of the proposed methods also involves one additional parameter tuning (r). In the case where two graphs are employed, an additional parameter a could also be fine-tuned if required.

Finally, as has been described in subsections 2.2 and 2.3, the proposed methods exploit a generic graph description of the form:

$$\tilde{\mathbf{S}} = \mathbf{\Phi}(\mathbf{D} - \mathbf{A})\mathbf{\Phi}^T + r\mathbf{I}, \tag{42}$$

where $[\mathbf{D}]_{ii} = \sum_{i \neq j}[\mathbf{A}]_{ij},\ i = 1, \ldots, N$ and $[\mathbf{D}]_{ij} = 0,\ i \neq j$. This fact allows us to exploit any type of graph designed (or that will be designed) within the Graph Embedding approach (as described in subsection 2.1). In addition, it allows us to exploit problem-related information through the creation of appropriate graph weights $\mathbf{A}$ (which may be even rule-based or hand-crafted, i.e. provided by a human expert). An example of data relationships that can be obtained by rule-based methods is the use of spatial video frame and temporal video information for face recognition. In that case, facial images belonging to the same person (determined by applying a face tracking technique) can be enforced to be similar by assigning a high graph weight value, while facial images appearing in the same video frame (and thus it is known that they belong to different persons) can be enforced to be dissimilar by assigning a low graph weight value. Such information can be of a high value, since in this way we can establish connections between frontal and side views of a person's face (e.g. if the person is depicted with a frontal view and turns to a side view in the succeeding frames), which would not be possible by exploiting distance-based criteria. In addition, very similar faces appearing in the same frame can be better distinguished.

3. Experiments

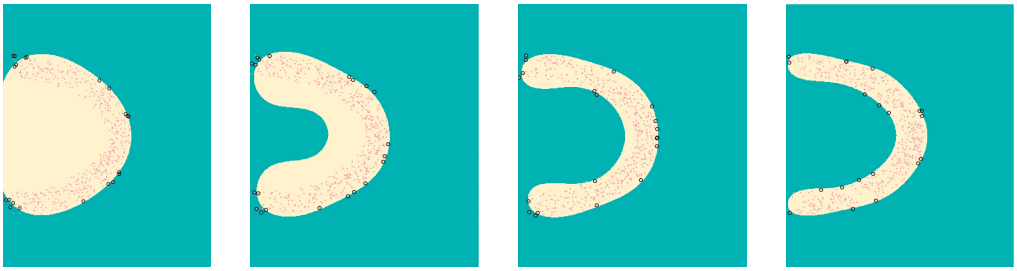
In this Section, we present experiments conducted in order to evaluate the performance of the proposed GE-OCSVM and GE-SVDD methods. We perform qualitative evaluation of the proposed methods in a 2-dimensional toy dataset and demonstrate the effects of the introduced parameters in Subsection 3.1. In order to perform quantitative evaluation, we applied the proposed methods in media data classification problems. The addressed media data classification problems include face recognition, human action recognition and video summarization, described in Subsections 3.3, 3.4 and 3.5, respectively. Moreover, we have applied the proposed method in well known standard one-class classification problems, including publicly available medical diagnostic problems and sonar signal classification, as described in Subsection 3.6. Information regarding the experimental protocol and parameter settings are described in Subsection 3.2.

3.1. Qualitative evaluation of different regularization terms

In order to demonstrate the effect of the various regularization choices that can be employed by the proposed methods, we have employed a toy dataset. In Figure 1, we illustrate the decision functions of the GE-OC-SVM, regularized by the total scatter of the training data. We demonstrate different settings of the parameter r. Decreasing the parameter r increases the effect of regularization in the obtained test space. More specifically, by decreasing r, the support vectors tend to be adjusted to low variance directions. That affects the shape of the resulting test space, so that it resembles the distribution of the training patterns. In this specific dataset, decreasing the value of parameter r, results in enclosing the training data in tighter boundaries. However, in the general case, decreasing the value of

parameter r too much may lead in over-fitted models.

In Figure 2, we demonstrate the decision functions obtained by applying the GE-OC-SVM using subclass information. Using this variant of the proposed GE-OC-SVM, the overall class distribution is considered along with the within-class data distributions of subclasses formed inside the training class. A different number of subclasses can be determined within the target class by applying a clustering approach [27]. We have determined 2, 3, 5 and 10 subclasses within the employed toy dataset. When a small number of subclasses is used, support vectors tend be placed around the total class distribution. When a large number of subclasses is used, the classification boundary is tightened to locations of subclass boundaries. This is useful when the training data form subclasses and the directions corresponding to the minimum variance are not appropriate to model the training data distribution.



(a) ℓ=0 (b) ℓ=-1 (c) ℓ=-2 (d) ℓ=-3

Figure 1: Demonstrating the effect of the parameter r of GE-OC-SVM, for different values of $r = 10^{\ell}$, by employing the total variance of the class, using $\nu = 0.05$. As can be seen in (a),(b),(c),(d), decreasing r, the support vectors tend to be adjusted to low variance directions. Thus, the derived test space follows the training data distribution.

(a) Z=2 (b) Z=3 (c) Z=5 (d) Z=10

Figure 2: Demonstrating the effect of increasing the number of subclasses GE-OC-SVM, by employing the within-subclass scatter, for a different number of subclasses Z. As can be seen in (a), (b), (c), (d), increasing Z increases the total number of support vectors. Moreover, it enforces the support vectors to be lying around subclasses formed inside the class, as shown especially in (d). The rest of the parameters were set $\nu = 0.05$ and $r = 0.01$, respectively. In this case, the obtained test space is adjusted to enclose groups of training data, as well as the data distribution.

3.2. Experimental protocol

All experiments were conducted on a Windows workstation featuring 32GB of RAM, using a MATLAB implementation. In all our experiments, we have applied the proposed method along with standard OCC methods. The competing methods include the proposed GE-OCSVM and GE-SVDD methods, the standard One-class Support Vector Machines (OC-SVM) [1], the standard Support Vector Data Description (SVDD) [2], the Laplacian One-Class Support Vector Machines (LAP-OC-SVM) [3] (in its one-class classification case), the Kernel Principal Component Analysis for Novelty Detection (KPCS) method [4] and the Kernel Null Space Method for Novelty Detection (KNFST) [5].

The best parameter values for each completing method were determined by using a set of parameter options and following a grid search strategy. The best set of parameters were automatically chosen by applying the n-fold cross validation

procedure, described in the following subsections.

For the proposed GE-OC-SVM method, we have tuned the parameter ν which controls the errors allowed in the optimization process and essentially controls the number of the resulting support vectors. Low values of ν limit the number of the resulting support vectors. We have chosen the parameter ν from a set of values $l = \{0.001, 0.01, 0.05, 0.1, 0.2, 0.3, 0.5\}$. Moreover, the kernel deformation parameter r was set to different orders of a magnitude, such that $r = 10^{\ell}$, where $\ell = -4, \ldots, 4$. In a similar fashion, we have tuned the corresponding parameters c and r of the proposed GE-SVDD. The parameter c was set to $c = 1/Nl$, and $r = 10^{\ell}$. Since GE-OC-SVM and GE-SVDD methods can exploit several graph types, we have employed four graph types in the conducted experiments, as well as a combination of two graph types. The exploited graph type is denoted with the respective acronym, as in GE-OC-SVM-X and GE-SVDD-X, where X can be equal to:

- ST, which denotes the use of the total scatter of the class.
- SW, which denotes the use of the within-subclass scatter, formed by $Z = \{2, 3, 5\}$ subclasses.
- FC, which denotes the use of the global geometric data relationships described by a fully connected weighted graph.
- KNN, which denotes the use of local geometric data relationships described in a kNN graph, where $k = \{5, 7\}$.
- SW-KNN, denotes the use of a combination of SW and KNN graphs.

For the remaining competing methods, the best parameters were determined as follows:

- For OC-SVM and SVDD, since they are special cases of the proposed methods, we employed the same set of parameters as in the proposed GE-OC-SVM and GE-SVDD.
- For LAP-OC-SVM we have employed the same set of parameters as in the GE-OC-SVM-KNN case.
- For KPCS, we have set the parameter p equal to $\{0.90, 0.95, 0.98\}$, where p is the PCA energy preserved. We have also set the parameter $N = \{0.001, 0.01, 0.05, 0.1, 0.2, 0.3, 0.5\}$, where N is a parameter denoting the importance of the reconstruction error, in the classification process.
- In KNFST we have set the reconstruction error importance parameter $N = \{0.001, 0.01, 0.05, 0.1, 0.2, 0.3, 0.5\}$.

In order to evaluate the performance of the methods we have employed the g-mean metric [30], which incorporates both precision and recall measurements as follows:

$$g = \sqrt{prec \times rec}. \tag{43}$$

G-mean has been designed for binary (imbalanced) classification problems, thus, is more suitable for our experiments, when compared to other metrics, e.g. classification rate.

3.3. Experiments in Face recognition

In order to evaluate the performance of the proposed methods in face recognition problems, we have employed the AR Face Database [31], the Yale Face

Database B [32] as well as the Public Figure and Labeled Faces in the Wild (PubFig83+LFW) [33] face recognition datasets.

In our first set of experiments, we have employed the AR and Yale datasets, which contain 2600 and 2432 frontal facial images from 100 and 38 subjects, respectively. We have employed the normalized pixel luminosities as feature vectors. To this end, the facial images were downsized to 40×30, and vectorized to produce 1200-dimensional vectors. We have performed the $5-$fold cross validation procedure, where we have split each dataset in 5 sets, mutually exclusive. For each fold, we created a number of binary problems, equal to the number of subjects, which is 100 for the AR and 38 for the Yale dataset, respectively. For each fold, we have trained the classifiers by employing the positive training samples belonging to the target class from 4 of the 5 sets, and tested on the remaining set, using all classes. This procedure was repeated five times, each for a test fold. Finally we report the average obtained g-mean metrics for all target classes.

In our second set of experiments, we employed the PubFig83+LFW dataset. We have employed the feature vectors (Histogram of Oriented Gradients, Local Binary Patterns, and Gabor wavelet features, reduced to 2048 dimensions with PCA), which were extracted from $13,002$ facial images representing 83 individuals from PubFig83, divided into 2/3 training (8720 faces) and 1/3 testing set ($4,282$ faces), as well as $12,066$ images representing over $5,000$ faces which were used as a distractor set from LFW. We have employed the first 1536 dimensions from the 2048, as suggested in [33]. For each of the 83 individuals, we have employed the training images for each class and tested on the respective test set of this class, as well the 500 first images of the distractor set.

In Table 2, we report the performance obtained for all subjects (all target

classes), in AR, Yale and PubFig83 + LFW datasets, respectively. In every case, all proposed GE-OC-SVM and GE-SVDD variants outperformed the standard OC-SVM and SVDD, respectively. When a single graph is employed, the best performance was reported by employing the proposed GE-OC-SVM-SW variant of the proposed method, which makes it suitable for face recognition applications. The best obtained performance in AR and Yale datasets, was reported when SW and KNN graphs were employed at the same time (GE-OC-SVM-SW-KNN).

3.4. Experiments in human action recognition

In this section, we present the experiments conducted in Human Action Recognition problems. For our experiments, we have employed the i3DPost multi-view action database [34], the IMPART Multi-modal/Multi-view Dataset [35], as well as the Hollywood2 [36] and Hollywood3D [37] publicly available datasets. The i3DPost dataset contains 512 high-resolution (1080×1920 pixel) videos depicting eight human actors performing eight activities. The database camera setup consists of eight cameras placed in the perimeter of a ring at a height of 2 meters above the studio floor. The IMPART dataset consists of a multi-camera outdoor setup, which consists of 14 fixed cameras placed around each subject, where each subject is performing 12 actions. In order to automatically create short video segments depicting distinct human activities from all cameras, we have employed a temporal video segmentation algorithm [38]. The Hollywood2 dataset consists of 810 training and 884 test video segments, of 12 activities. Finally, the Hollywood3D dataset consists of 359 train and 307 test stereoscopic video segments depicting 14 actions. In our experiments, we have employed only the right video channel.

In order to obtain vectorial video representations for each video segment de-

Table 2: Average g-means performance in Face recognition datasets

	AR	YALE	PubFig83+ LFW
OC-SVM [1]	71.41	63.71	76.02
SVDD [2]	70.39	63.42	76.55
LAP OC-SVM [3]	75.07	71.64	76.95
KPCS [4]	73.23	68.61	28.77
KNFST [5]	38.18	39.25	56.50
GE-OC-SVM-FC	74.81	74.93	77.52
GE-OC-SVM-ST	75.01	75.11	77.34
GE-OC-SVM-SW	83.23	81.00	**78.23**
GE-OC-SVM-KNN	74.79	67.98	77.02
GE-SVDD-FC	71.86	64.90	77.20
GE-SVDD-ST	71.88	65.13	77.45
GE-SVDD-SW	73.40	66.59	77.94
GE-SVDD-KNN	74.15	68.07	77.15
GE-OC-SVM-SW-KNN	**84.59**	**81.25**	76.94

picting one activity, we have employed the dense trajectory-based video description [39]. This video description calculates five descriptor types, namely the Histogram of Oriented Gradients (HOG), Histogram of Optical Flow (HOF), Motion Boundary Histogram along direction x (MBHx), Motion Boundary Histogram along direction y (MBHy) and the normalized trajectory coordinates (Traj), on the trajectories of densely-sampled video frame interest points that are tracked for a number of consecutive video frames (7 frames are used in our experiments). The five descriptors are calculated on the trajectory of each video frame interest point. We haved employed these video segment descriptions in order to obtain five video segment representations by using the Bag-of-Words model [40]. Thus, by following this process, each video segment was represented by 5 vectors, i.e. $\mathbf{x}_i^d,\ d = 1, \ldots, 5$. In order to fuse the information described in different video representations, we have combined the video segment representations with kernel methods, as in [39]. That is, we have employed the RBF kernel function, combining different descriptor types using a multi-channel approach [41]:

$$k(\mathcal{X}_i, \mathcal{X}_j) = exp\left(-\frac{1}{d}\sum_d \frac{\|\mathbf{x}_i^d - \mathbf{x}_j^d\|_2^2}{2\sigma_d^2}\right), \tag{44}$$

where σ_d is a parameter scaling the Euclidean distance between $\mathbf{x}_i^d$ and $\mathbf{x}_j^d$. In our experiments, we set the value of σ_d proportional to the mean Euclidean distance between the $\mathbf{x}_i^d, i = 1, \ldots, N$, which is the natural scaling factor for the Euclidean distances for each descriptor type on each dataset. After calculating the kernel matrices for the training and test samples, we employed them in each classification problem.

In the i3DPost and IMPART datasets, we have employed a 3-fold cross validation procedure, where we have split the datasets in 3 sets, mutually exclusive.

Each set included videos depicting all activities. We have employed the videos depicting each distinct activity from two sets in order to train the classifiers, and tested on the remaining one. For each activity, we have obtained g-mean metric. This procedure was repeated for all activities, and repeated 3 times for each fold. In the Hollywood2 and Hollywood 3D datasets, we employed the standard train and test videos, provided by the authors of [36, 37]. The average g-mean metrics obtained for all activities between the folds is depicted in Table 3.

It all cases, every variant of the proposed GE-OC-SVM outperformed the standard OC-SVM, as in face recognition. The proposed GE-SVDD outperformed the standard SVDD, in most cases. The GE-SVDD-KNN variant outperformed other GE-SVDD variants. LAP-OC-SVM outperformed the proposed methods in IMPART dataset. In all other cases, the maximum performance was obtained with the proposed GE-OC-SVM-SW-KNN method.

3.5. Experiments on video summarization

In this section, we present the experiments conducted in order to evaluate the performance of the proposed methods in the video summarization scenario. To this end, we have created a dataset where we seek the most interesting parts of a movie, based on similarity to generic movie trailers, namely the *IMPART-AUTH Movie Trailer Dataset*. This dataset has been created to provide a movie summarization scenario, since other publicly available datasets specially created for video summarization (e.g., [42]) provide simplistic or unrealistic video summarization scenarios. In the case of movies, video shots appearing in movie trailers are good examples of salient video segments, since they have been specially edited, in order to catch the viewer attention and, at the same time, to describe the movie plot. In order to train the classifiers, we have employed thirty movie trailers. In order to

Table 3: Average g-means rates in Human action recognition datasets

	IMPART	i3DPost	Hollywood2	Hollywood3D
OC-SVM [1]	61.45	74.53	58.54	55.90
SVDD [2]	61.55	74.19	58.29	56.15
LAP OC-SVM [3]	**78.09**	84.87	61.91	62.48
KPCS [4]	43.97	78.30	55.99	28.98
KNFST [5]	69.61	77.47	53.87	55.28
GE-OC-SVM-FC	64.94	76.09	62.53	58.22
GE-OC-SVM-ST	64.69	75.86	62.70	58.02
GE-OC-SVM-SW	65.35	77.56	64.49	59.77
GE-OC-SVM-KNN	71.70	86.52	59.66	59.82
GE-SVDD-FC	62.22	73.58	60.12	58.08
GE-SVDD-ST	62.70	74.43	59.84	58.61
GE-SVDD-SW	63.10	74.86	61.19	58.72
GE-SVDD-KNN	69.67	82.65	59.37	57.98
GE-OC-SVM-SW-KNN	77.91	**87.54**	**66.88**	**64.29**

test the summarization performance, we have employed three full-length movies. The employed movies genres was action, adventure and drama, respectively. The movie trailers belong to action, adventure, comedy, thriller and drama categories. We would like to employ a one-class classification model in order to retrieve the video segments that were employed to form the real movie trailer.

In order to segment the videos into short video segments, we have employed a shot-cut detection algorithm [43]. This procedure resulted in 2788 video segments of the movie trailers, 1961 for the first, 2099 for the second and 2687 for the third movie. The ground truth contained information on whether a video segment of a test movie was employed in order to create its own movie trailer. In the annotation process we have observed that, in some cases, video segments that appear in the movie trailers are shorter or contain some scenes that do not appear in the respective movies. We have annotated the longer video segments contained in movies as salient, creating three summarization scenarios for each movie, containing a total of 6310 non salient and 437 salient salient video segments for test. Since saliency in this sense can be related to human activity, we have employed the Dense trajectories video description [39], which is the same as the one employed in Human action recognition problems.

We have employed the 2788 video segments of the movie trailers, in order to train the classifiers and the 6747 video segments to test the summarization performance. We should note here that the trailers of the three (test) movies were not included in the training set. Experimental results are shown in Table 4. We report the obtained g-mean, F-measure, precision and recall metrics. All variants of the proposed GE-OC-SVM outperformed the standard OC-SVM, as well as the LAP OC-SVM. The proposed GE-SVDD outperformed the standard SVDD, in most

cases.

Table 4: Performance in IMPART-AUTH Movie Trailer Dataset.

	G-mean	F-measure	precision	recall
OC-SVM [1]	51.54	51.53	52.74	50.36
SVDD [2]	49.06	48.61	42.82	56.20
LAP OC-SVM [3]	48.99	48.61	43.26	55.47
KPCS [4]	50.84	50.83	52.08	49.64
KNFST [5]	44.00	37.66	**78.02**	24.82
GE-OC-SVM-FC	53.10	53.02	55.98	50.36
GE-OC-SVM-ST	54.14	54.03	50.82	57.66
GE-OC-SVM-SW	53.81	53.77	55.87	51.82
GE-OC-SVM-KNN	**58.81**	**58.81**	58.50	**59.12**
GE-SVDD-FC	48.49	47.96	41.83	56.20
GE-SVDD-ST	51.33	51.30	49.45	53.28
GE-SVDD-SW	50.45	50.23	45.89	55.47
GE-SVDD-KNN	55.52	55.51	56.30	54.74
GE-OC-SVM-SW-KNN	58.56	58.56	58.00	**59.12**

3.6. Experiments on standard OCC problems

In order to further evaluate the performance of the proposed GE-OC-SVM and GE-SVDD methods, we have also employed standard One-Class Classification problems, namely the Arrythmia (ARR), Breast Benign (BB), Breast Malignant (BM), Diabetes (DB), Heart (HRT), Liver (LVR), Sonar Mines (SM), Sonar Rocks (SR) and Thyroid (THR) datasets. The above mentioned datasets are publicly

available in the UCI repository [44]. One-class versions can also be downloaded from Pattern Recognition Laboratory, Netherlands, using the DDtools library [25], which is the case in the present paper.

Each dataset consisted of a binary classification problem, containing positive and negative examples. For each dataset, we have performed the 5-fold cross validation procedure, where 80% each set of each dataset was employed for training purposes and 20% for test. We have trained the classifiers by employing the positive examples in each dataset. The reported performance is the average performance obtained between each fold. Experimental results are shown in Table 5. Each column depicts the best obtained performance for each method in each dataset. The last column contains the average obtained performance for all the datasets. As can be seen, in almost every dataset, all versions of the proposed GE-OC-SVM and GE-SVDD perform consistently better than the standard OC-SVM and SVDD, respectively, having an average $5 - 7\%$ gain, depending on the exploited graph type. The proposed methods outperformed the competition, in most of the cases. Finally, the performance of the proposed methods is further enhanced when the combination of SW and KNN graphs is employed.

4. Conclusion

In this paper, we have described a generic One-Class classification framework that exploits geometric data relationships in the OC-SVM and SVDD optimization processes. We have shown that the adoption of geometric class information improves the target class modeling, by acting as a regularization term. Moreover, the proposed method can be applied using existing OC-SVM and SVDD implementations, in several One-Class classification problems.

Table 5: G-means Rates for standard OCC problems.

	ARR	BB	BM	DB	HRT	LVR	SM	SR	THR	AVG
OC-SVM [1]	73.14	95.67	90.99	55.05	57.51	55.53	60.62	59.45	52.91	66.76
SVDD [2]	72.60	95.55	92.09	25.08	55.12	48.47	60.62	57.68	50.38	61.96
LAP OC-SVM [3]	39.48	93.63	24.33	57.35	55.63	56.10	52.91	56.96	65.87	55.81
KPCS [4]	54.33	80.02	23.56	55.76	50.59	50.17	72.68	55.15	62.02	56.03
KNFST [5]	72.29	96.57	80.58	45.68	47.04	42.97	54.11	48.64	50.56	59.83
GE-OC-SVM-FC	73.14	96.22	92.37	64.40	63.94	60.68	65.66	60.34	67.29	71.56
GE-OC-SVM-ST	73.21	96.22	91.89	59.22	64.23	62.80	66.92	60.86	64.78	71.13
GE-OC-SVM-SW	74.17	96.22	92.12	61.61	65.07	61.66	71.01	61.53	67.63	72.34
GE-OC-SVM-KNN	73.14	97.74	94.15	55.59	59.89	**64.31**	60.62	59.45	52.91	68.65
GE-SVDD-FC	73.73	95.55	95.98	49.99	58.40	56.76	61.09	59.45	53.16	67.12
GE-SVDD-ST	72.60	96.12	96.43	49.59	59.89	55.47	60.62	63.54	52.95	67.47
GE-SVDD-SW	72.60	96.68	96.27	56.91	62.98	59.90	61.09	61.84	54.04	69.15
GE-SVDD-KNN	72.60	96.68	96.27	44.41	54.43	59.06	60.62	59.45	52.91	66.27
GE-OC-SVM-SW-KNN	**77.27**	**98.49**	**97.42**	**65.02**	**65.44**	61.45	**73.85**	**74.06**	**77.71**	**76.74**

Future work can include inducing additional manifold learning/discriminating criteria in the optimization process, as well as linear combinations of multiple graph types. Our work could also be extended in the scope of automatic determination of training parameters introduced by our methods.

Acknowledgement

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement number 316564 (IMPART).

Appendix A. Graph Embedded One-Class SVM solution

Here we present the auxiliary steps performed to derive in the final GE-OC-SVM solution (22). The dual GE-OC-SVM optimization problem is given by:

$$\mathcal{L} = \frac{1}{2}\boldsymbol{\lambda}^T\mathbf{K}(\mathbf{KLK} + r\mathbf{K})^{-1}\mathbf{K}\boldsymbol{\lambda}. \tag{A.1}$$

The solution is similar with the standard OC-SVM if we replace $\tilde{\mathbf{K}} = \mathbf{K}(\mathbf{KLK} + r\mathbf{K})^{-1}\mathbf{K}$. Using the following steps, $\tilde{\mathbf{K}}$ takes the form:

$$\begin{aligned} \tilde{\mathbf{K}} &= \mathbf{K}\left[\frac{1}{r}\mathbf{K}^{-1}\frac{1}{r^2}(\mathbf{L}^{-1} + \frac{1}{r}\mathbf{K})^{-1}\right]\mathbf{K} \\ \tilde{\mathbf{K}} &= \frac{1}{r}\left[\mathbf{I} - (\mathbf{L} + r\mathbf{K}^{-1})^{-1}\mathbf{L}\right]\mathbf{K}. \end{aligned} \tag{A.2}$$

Then, the dual GE-OC-SVM optimization function is given by:

$$\mathcal{L} = \frac{1}{2r}\boldsymbol{\lambda}^T\tilde{\mathbf{K}}\boldsymbol{\lambda}, \tag{A.3}$$

which is of the same form as the standard OC-SVM solution. Thus, any standard OC-SVM implementation, e.g., [24], can be exploited to find the solution of the GE-OC-SVM optimization problem, using the deformed kernel $\tilde{\mathbf{K}}$.

Appendix B. Graph Embedded SVDD solution

In this Appendix we present the auxiliary steps performed in order to compute the final solution of GE-SVDD. The Lagrangian function obtained after replacing equations (30), (31) and (32) in (29) is as follows:

$$\mathcal{L} = \sum_{i=1}^{N}\lambda_i\phi(\mathbf{x}_i)^T\tilde{\mathbf{S}}^{-1}\phi(\mathbf{x}_i) - \sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_i\lambda_j\phi(\mathbf{x}_i)^T\tilde{\mathbf{S}}^{-1}\phi(\mathbf{x}_j), \tag{B.1}$$

where λ_i are the Lagrange multipliers and $\tilde{\mathbf{S}}$ is defined in (9). By using the Woodbury identity, we have:

$$\tilde{\mathbf{S}}^{-1} = \frac{1}{r}\mathbf{I} - \frac{1}{r^2}\mathbf{\Phi}(\mathbf{L^{-1}} + \frac{1}{r}\mathbf{K})^{-1}\mathbf{\Phi}^T. \tag{B.2}$$

Replacing (B.2) in (B.1) we obtain:

$$\begin{aligned}
\mathcal{L} &= \sum_{i=1}^{N} \lambda_i \boldsymbol{\phi}_i^T \left(\frac{1}{r}\mathbf{I} - \frac{1}{r^2}\mathbf{\Phi}(\mathbf{L^{-1}} + \frac{1}{r}\mathbf{K})^{-1}\mathbf{\Phi}^T \right) \boldsymbol{\phi}_i - \\
&- \sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i \lambda_j \boldsymbol{\phi}_i^T \left(\frac{1}{r}\mathbf{I} - \frac{1}{r^2}\mathbf{\Phi}(\mathbf{L^{-1}} + \frac{1}{r}\mathbf{K})^{-1}\mathbf{\Phi}^T \right) \boldsymbol{\phi}_j, \\
\mathcal{L} &= \sum_{i=1}^{N} \lambda_i \left(\frac{1}{r}k_{ii} - \frac{1}{r^2}\mathbf{k}_i^T(\mathbf{L^{-1}} + \frac{1}{r}\mathbf{K})^{-1}\mathbf{k}_i \right) - \\
&- \sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i \lambda_j \left(\frac{1}{r}k_{ij} - \frac{1}{r^2}\mathbf{k}_i^T(\mathbf{L^{-1}} + \frac{1}{r}\mathbf{K})^{-1}\mathbf{k}_j \right), \\
\mathcal{L} &= \sum_{i=1}^{N} \lambda_i \left(\frac{1}{r}k_{ii} - \frac{1}{r^2}\mathbf{k}_i^T(\mathbf{L^{-1}} + \frac{1}{r}\mathbf{K})^{-1}\mathbf{k}_i \right) - \\
&- \boldsymbol{\lambda}^T \left(\frac{1}{r}\mathbf{K} - \frac{1}{r^2}\mathbf{K}(\mathbf{L^{-1}} + \frac{1}{r}\mathbf{K})^{-1}\mathbf{K} \right) \boldsymbol{\lambda}.
\end{aligned} \tag{B.3}$$

We observe that the solution of the GE-SVDD is equivalent to solution of the standard SVDD, in a different space. In order to exploit standard SVDD implementations, we can employ the following matrix in the standard SVDD optimization problem:

$$\begin{aligned}
\tilde{\mathbf{K}} &= \frac{1}{r}\mathbf{K} - \frac{1}{r^2}\mathbf{K}\left(\mathbf{L}^{-1} + \frac{1}{r}\mathbf{K} \right)^{-1}\mathbf{K}, \\
\tilde{\mathbf{K}} &= \frac{1}{r}\mathbf{K} - \frac{1}{r^2}\mathbf{K}\left[r\mathbf{K}^{-1}(\mathbf{L} + r\mathbf{K}^{-1})^{-1}\mathbf{L} \right]\mathbf{K}, \\
\tilde{\mathbf{K}} &= \frac{1}{r}\mathbf{K} - \frac{1}{r}(\mathbf{L} + r\mathbf{K}^{-1})^{-1}\mathbf{L}\mathbf{K}, \\
\tilde{\mathbf{K}} &= \frac{1}{r}\left[\mathbf{I} - (\mathbf{L} + r\mathbf{K}^{-1})^{-1}\mathbf{L} \right]\mathbf{K}.
\end{aligned} \tag{B.4}$$

Thus, the derived space for GE-SVDD (B.4) is equivalent to the one GE-OC-SVM, found in (A.2).

References

[1] B. Scholkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, R. C. Williamson, Estimating the support of a high-dimensional distribution, Neural computation 13 (7) (2001) 1443–1471.

[2] D. M. Tax, R. P. Duin, Support vector data description, Machine learning 54 (1) (2004) 45–66.

[3] J. Muñoz-Marí, F. Bovolo, L. Gómez-Chova, L. Bruzzone, G. Camp-Valls, Semisupervised one-class support vector machines for classification of remote sensing data, IEEE Transactions on Geoscience and Remote Sensing 48 (8) (2010) 3188–3197.

[4] H. Hoffmann, Kernel pca for novelty detection, Pattern Recognition 40 (3) (2007) 863–874.

[5] P. Bodesheim, A. Freytag, E. Rodner, M. Kemmler, J. Denzler, Kernel null space methods for novelty detection, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2013) 3374–3381.

[6] V. Mygdalis, A. Iosifidis, A. Tefas, I. Pitas, Video summarization based on subclass support vector data description, IEEE Symposium Series on Computational Intelligence (SSCI), IEEE Symposium on Computational Intelligence for Engineering Solutions (CIES) (2014) 183–187.

[7] V. Mygdalis, A. Iosifidis, A. Tefas, I. Pitas, Exploiting subclass information in one-class support vector machine for video summarization, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2015) 2259–2263.

[8] B. Cyganek, One-class support vector ensembles for image segmentation and classification, Journal of Mathematical Imaging and Vision 42 (2-3) (2012) 103–117.

[9] M. A. Pimentel, D. A. Clifton, L. Clifton, L. Tarassenko, A review of novelty detection, Signal Processing 99 (2014) 215–249.

[10] N. M. Khan, R. Ksantini, I. S. Ahmad, L. Guan, Covariance-guided one-class support vector machine, Pattern Recognition 47 (6) (2014) 2165–2177.

[11] D. Tao, X. Li, X. Wu, S. J. Maybank, Geometric mean for subspace selection, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2) (2009) 260–274.

[12] D. Tao, X. Li, X. Wu, S. J. Maybank, General tensor discriminant analysis and gabor features for gait recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (10) (2007) 1700–1715.

[13] J. Yu, Y. Rui, Y. Y. Tang, D. Tao, High-order distance-based multiview stochastic learning in image classification, IEEE Transactions on Cybernetics 44 (12) (2014) 2431–2442.

[14] J. Yu, Y. Rui, D. Tao, Click prediction for web image reranking using multimodal sparse coding, IEEE Transactions on Image Processing 23 (5) (2014) 2019–2032.

[15] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, S. Lin, Graph embedding and extensions: a general framework for dimensionality reduction, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (1) (2007) 40–51.

[16] J. Yu, D. Tao, Y. Rui, J. Cheng, Pairwise constraints based multiview features fusion for scene classification, Pattern Recognition 46 (2) (2013) 483–496.

[17] G. Arvanitidis, A. Tefas, Exploiting graph embedding in support vector machines, IEEE International Workshop on Machine Learning for Signal Processing (MLSP) (2012) 1–6.

[18] A. Iosifidis, A. Tefas, I. Pitas, Graph embedded extreme learning machine, IEEE Transactions on Cybernetics 46 (1) (2016) 311–324.

[19] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: A geometric framework for learning from labeled and unlabeled examples, The Journal of Machine Learning Research 7 (2006) 2399–2434.

[20] S. Zafeiriou, N. Laskaris, On the improvement of support vector techniques for clustering by means of whitening transform, IEEE Signal Processing Letters 15 (2008) 198–201.

[21] A. Iosifidis, A. Tefas, I. Pitas, Minimum class variance extreme learning machine for human action recognition, IEEE Transactions on Circuits and Systems for Video Technology 23 (11) (2013) 1968–1979.

[22] B. Schölkopf, A. J. Smola, Learning with kernels: Support vector machines, regularization, optimization, and beyond.

[23] T. Le, D. Tran, W. Ma, D. Sharma, A unified model for support vector machine and support vector data description, International Joint Conference on Neural Networks (IJCNN) (2012) 1–8.

[24] C. C. Chang, C. J. Lin, LIBSVM: A library for support vector machines, ACM Transactions on Intelligent Systems and Technology 2 (3) (2011) 1–27.

[25] D. Tax, Ddtools, the data description toolbox for matlabVersion 2.1.2.

[26] A. Maronidis, A. Tefas, I. Pitas, Subclass graph embedding and a marginal fisher analysis paradigm, Pattern Recognition 48 (12) (2015) 4024–4035.

[27] J. Shawe-Taylor, N. Cristianini, Kernel methods for pattern analysis.

[28] B. Schölkopf, R. Herbrich, A. J. Smola, A generalized representer theorem, Computational learning theory (2001) 416–426.

[29] H. Yu, Single-class classification with mapping convergence, Machine Learning 61 (1-3) (2005) 49–69.

[30] M. Kubat, R. C. Holte, S. Matwin, Machine learning for the detection of oil spills in satellite radar images, Machine learning 30 (2-3) (1998) 195–215.

[31] A. M. Martinez, The ar face database, CVC Technical Report 24.

[32] A. S. Georghiades, P. N. Belhumeur, D. J. Kriegman, From few to many: Illumination cone models for face recognition under variable lighting and pose, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (6) (2001) 643–660.

[33] B. Becker, E. Ortiz, Evaluating open-universe face identification on the web, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2013) 904–911.

[34] N. Gkalelis, H. Kim, A. Hilton, N. Nikolaidis, I. Pitas, The i3dpost multi-view and 3d human action/interaction database, Conference for Visual Media Production (CVMP) (2009) 159–168.

[35] H. Kim, A. Hilton, Influence of colour and feature geometry on multi-modal 3d point clouds data registration, International Conference on 3D Vision (3DV) 1 (2014) 202–209.

[36] M. Marszalek, I. Laptev, C. Schmid, Actions in context, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2009) 2929–2936.

[37] S. Hadfield, R. Bowden, Hollywood 3d: Recognizing actions in 3d natural scenes, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2013) 3398–3405.

[38] N. Kourous, A. Iosifidis, A. Tefas, N. Nikolaidis, I. Pitas, Video characterization based on activity clustering, IEEE International Conference on Electrical and Computer Engineering (ICECE) (2014) 266–269.

[39] H. Wang, A. Kläser, C. Schmid, C.-L. Liu, Dense trajectories and motion boundary descriptors for action recognition, International journal of computer vision 103 (1) (2013) 60–79.

[40] A. Iosifidis, A. Tefas, I. Pitas, Discriminant bag of words based representation for human action recognition, Pattern Recognition Letters.

[41] J. Zhang, M. Marszalek, M. Lazebnik, C. Schmid, Local features and kernels for classification of texture and object categories: A comprehensive study, International Journal of Computer Vision 73 (2) (2007) 213–238.

[42] S. E. F. de Avila, A. P. B. Lopes, et al., Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method, Pattern Recognition Letters 32 (1) (2011) 56–68.

[43] Z. Černeková, I. Pitas, C. Nikou, Information theory-based shot cut/fade detection and video summarization, IEEE Transactions on Circuits and Systems for Video Technology 16 (1) (2006) 82–91.

[44] M. Lichman, UCI machine learning repository (2013).
URL `http://archive.ics.uci.edu/ml`