Exploiting multiplex data relationships in Support Vector Machines

Vasileios Mygdalis, Anastasios Tefas, Ioannis Pitas

Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece,
Author pre-print version, submitted to Elsevier Pattern Recognition.
Please cite the publisher maintained version in your work.
DOI : https://doi.org/10.1016/J.PATCOG.2018.07.032
Copyright: 2018 Elsevier Ltd.

Abstract

In this paper, a novel method for introducing multiplex data relationships to the SVM optimization process is presented. Different properties about the training data are encoded in graph structures, in the form of pairwise data relationships. Then, they are incorporated to the SVM optimization problem, as modified graph-regularized base kernels, each highlighting a different property about the training data. The contribution of each graph-regularized kernel to the SVM classification problem, is estimated automatically. Thereby, the solution of the proposed modified SVM optimization problem lies in a regularized space, where data similarity is expressed by a linear combination of multiple single-graph regularized kernels. The proposed method exploits and extends the findings of Multiple Kernel Learning and graph-based SVM method families. It is shown that the available kernel options for the former can be broadened, and the exhaustive parameter tuning for the latter can be eliminated. Moreover, both method families can be considered as special cases of the proposed formulation, hereafter. Our experimental evaluation in visual data classification problems denote the superiority of the proposed method. The obtained classification performance gains can be explained by the exploitation of multiplex data relationships, during the classifier optimization process.

Keywords: Multiplex data relationships, Support Vector Machine, Graph-based Regularization, Multiple Kernel Learning.

1. Introduction

Computer vision/visual analysis methods have found industrial applications in several areas such as in robotic systems e.g., unmanned aerial vehicles and virtual reality, and their growth over the past few years have been immense. Such visual analysis applications including face recognition, object recognition, human action recognition, human/object tracking and many other applications, are commonly addressed as classification problems [1, 2]. One of the most widely studied classification methods in visual analysis applications is the Support Vector Machines (SVM) classifier. SVM-based methods and extensions have been employed in mathematical/engineering problems including one-class and multiclass classification, regression and semi-supervised learning [3, 4, 5, 6]. In its simplest form, SVM learns from labeled data examples originating from two classes, the hyperplane that separates them with the maximum margin, at the training data input (or feature) space. After its first proposal, SVM has been extended to determine decision functions in feature spaces obtained by employing non-linear data mappings, where data similarity

is implicitly expressed by a kernel function. The explicit data mapping is not required to be known, if the adopted kernel function satisfies Mercer conditions [7]. Common practices for determining a feature space where SVM provides satisfactory performance to a given classification/regression problem, involve selecting a kernel function from a set of widely adopted kernel functions e.g., polynomial, sigmoid, Radial Basis Function (RBF), and thereby tuning the corresponding hyperparameters using e.g., cross validation, based on previous knowledge about the problem at hand. In every case, the performance of SVM heavily depends on the adopted kernel function choice, since the optimal solution for each problem might lie in unknown feature spaces.

In order to determine the optimal feature space for SVM operation, Multiple Kernel Learning (MKL) methods have been proposed. Their basic assumption is that the optimal underlying data mapping, i.e., the optimal kernel function, is a weighted combination (either linear or non-linear) of multiple kernel functions, the so-called basis kernels [8, 9, 10, 11]. The participation of each kernel to the optimal solution is determined by the kernel weights. The weights of the basis kernels are estimated in an automated fashion along with the SVM hyperplane, by an additional optimization procedure (e.g., single-step sequential optimization, two-step optimization). Standard MKL methods employ L_p or L_1 regularization in their optimization procedure, with the latter producing sparse solutions and the former providing fast convergence [12, 13]. Besides the important theoretical advancements of MKL methods, only few base kernel combinations have found to be successful in realistic applications, i.e., MKL methods method might suffer from overfitting issues or limited performance gains [11, 12, 13].

A different approach for improving classification performance, are methods that introduce additional optimization criteria to the standard SVM optimization problem, such as discriminant/manifold learning [6]. That is, alternative optimization problems have been proposed in order to determine SVM solutions in regularized spaces, expressed by a geometric transformation of the derived SVM hyperplane with the adopted criteria. For example, employing discriminant learning information e.g., within-class variance information [14], promotes hyperplanes that span along low data variance directions [15, 16]. Alternatively, methods initially proposed for semi-supervised learning, by integrating SVM and manifold learning [6], have shown that enhanced classification performance can be obtained for the supervised learning case as well, by exploiting $k-$Nearest Neighborhood (k-NN) graphs. Since advances in graph-theory allow several manifold/discriminant learning criteria to be expressed using graph-based representation [17], methods incorporating the undelying data geometry in the SVM optimization problem can be implemented through generic graph-based SVM methods [18, 19, 20]. The adoption of generic graph structures within the SVM optimization process, containing e.g., intrinsic (within-class), or between-class data relationships, promotes solutions that are less prone to over-fitting. The disadvantage of graph-based SVM methods is that deriving the optimal classification space requires the evaluation of different graph settings, as well as tuning the additional introduced hyperparameters.

In visual analysis applications, MKL and graph-based SVM methods have been succesfully employed over the past few years. Their success can be mainly attributed to the expoitation of the multimodal/multiplex structure of images and video data [21]. The multimodal/multiplex structrure can be related to spatial and temporal information, information extracted by multiple descriptor types, or even noise generated by camera movement, multiple viewing angles and illumination changes. In order to handle such information, adopting more than a single kernel [11], or adding more than a single graph [22, 23, 24, 25, 26], is beneficial to performance, since it enables more accurate representation of the underlying multiplex data relationships. Our work was inspired by the successful implementation of multiple graphs in related application scenarios, e.g., label propagation [22]. To this end, we have devised a method that introduces multiple graphs to the SVM optimization problem, by exploiting the intuitions of both MKL and graph-based SVM method families.

In this paper, a novel classification method that incorporates multiplex data relationships to the SVM

optimization process, is presented. Multiplex data relationships are encoded in the form of multiple graph structures, containing pairwise data relationships, each corresponding to a specific data property. We propose a modified SVM optimization problem, that incorporates this information to its optimization problem. As an effect, the generated SVM hyperplane is driven to directions where the most discriminant training data properties are highlighted. From our derivations, it is shown that the solution of the proposed optimization problem lies in a modified space, where data similarity is explicitly determined by a linear combination of graph-regularized kernel matrices. Moreover, it is proven that both Multiple Kernel Learning and Graph-based SVM method families method families can be formulated as special cases of the proposed method, hereafter. Finally, the proposed method exploits and extends the findings of Multiple Kernel Learning and graph-based SVM method families, by broadening the available kernel options for the former, and eliminating exhaustive parameter tuning for the latter.

2. Related Work

In this section, we overview the preliminary material required to introduce the proposed method. Section 2.1 contains the description of the generic MKL-SVM optimization problem and Section 2.2 contains an overview of the recently proposed Graph-Embedded Support Vector Machines, exploiting a single graph in its optimization problem for regularization purposes.

2.1. Multiple Kernel Learning Support Vector Machines

Let a set of labeled data $\mathcal{S} = \{\boldsymbol{x}_i, y_i\}, i = 1, \ldots, N$ sampled from $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} \in \mathbb{R}^D$ and $\mathcal{Y} \in \{-1, 1\}$, that is employed in order to train an SVM classifier. MKL-SVM methods optimize for implicitely determining the optimal feature space for solving the SVM optimization problem. Similarity in that space is reproduced by a linear or non-linear combination of multiple kernel functions [10, 13, 27, 28, 29, 30]. Let M mapping functions $\phi_m(\cdot) \mapsto \mathcal{H}^m, m = 1, \ldots, M$ that have been employed as base data mappings. Similarity in the respective spaces is reproduced by the associated base kernel function $\kappa_m(\cdot, \cdot) = \phi_m(\cdot)^T \phi_m(\cdot)$, and $\mathcal{H}^m$ is a Reproducing Kernel Hilbert Space (RKHS). Assuming M base kernels have been linearly combined, then the obtained space $\mathcal{H}$ is also a RKHS, reproduced by kernel $\kappa(\cdot, \cdot)$. Similarity in that space can be calculated explicitly by a weighted summation of the base kernels, as follows:

$$\kappa(\cdot, \cdot) = \sum_{m=1}^{M} \mu_m \kappa_m(\cdot, \cdot), \tag{1}$$

where κ_m is the $m-$th kernel function weighted by a parameter $\mu_m \geq 0$.

In order to learn the kernel weighting parameters μ_m and the optimal SVM hyperplane at the same time, the MKL-SVM optimization problem is formed as a max-min optimization problem:

$$\max_{\boldsymbol{\alpha}} \ \min_{\boldsymbol{\mu}} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \sum_{m=1}^{M} \mu_m \kappa_m(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{2}$$

$$\text{s.t.} \quad 0 \leq \alpha_i \leq c \ \text{ and } \ \sum_{m=1}^{M} \mu_m^p = 1,$$

where $\boldsymbol{a}$ is the support vector coefficient vector and $p \geq 1$ is a parameter that affects the sparsity of the obtained kernel weighting parameters. The above defined optimization problem can be solved sequentially or in an iterative manner, keeping $\boldsymbol{a}$ or $\boldsymbol{\mu}$ as constants in the respective optimization steps. Assuming that the kernel weighting parameters $\boldsymbol{\mu}$ have been determined, then $\boldsymbol{K} = \sum_{m=1}^{M} \mu_m \boldsymbol{K}_m$ is the kernel matrix that can be employed for solving the standard SVM classification problem. According to Representer Theorem [7], the relevant SVM hyperplane $\boldsymbol{w} = \boldsymbol{\Phi a}$ that lies in the RKHS $\mathcal{H}$, can be reconstructed by the determined support vector coefficient vector $\boldsymbol{a}$ and the arbitrary training data representations $\boldsymbol{\Phi} \in \mathcal{H}$. Data similarity in that space can only be reproduced by the base kernel combination, since the kernel $\boldsymbol{K}$ cannot be calculated, otherwise.

After training the classifier, a test sample $\boldsymbol{x}$ is classified to the positive or negative training class, according to the outputs of the following decision function:

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} y_i \alpha_i \sum_{m=1}^{M} \mu_m \kappa_m(\boldsymbol{x}_i, \boldsymbol{x}) + b, \qquad (3)$$

where b is the standard SVM bias term. Finally, the test sample is classified to the positive class if $sign(f(\boldsymbol{x})) \geq 0$ or the negative class, otherwise.

2.2. Support Vector Machines exploiting geometric data relationships

Graph-based SVM methods exploit data relationships expressed by a single graph in the SVM optimization problem [18, 20]. To this end, it is assumed that the training data $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ have been embedded in an undirected weighted graph $\mathcal{G} = \{\mathcal{X}, \mathbf{W}\}$, where $\boldsymbol{W} \in \mathbb{R}^{N \times N}$ is the graph weight matrix. It should be noted that non-linear data relationships might be expressed as well, by employing the explicit data mappings in a feature space i.e., $\mathcal{X} = \{\phi(\boldsymbol{x}_1), \ldots, \phi(\boldsymbol{x}_N)\}$, where $\phi(\cdot) : \mathbb{R}^D \mapsto \mathcal{H}$ is mapping function. In either case, the matrix $\boldsymbol{S}$ can be employed to preserve data relationships expressed by $\mathcal{G}$, in the feature space $\mathcal{H}$. The definition of $\boldsymbol{S}$ is the following:

$$\begin{aligned} \boldsymbol{S} =& \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} W_{ij} \left(\phi(\boldsymbol{x}_i) - \phi(\boldsymbol{x}_j)\right) \left(\phi(\boldsymbol{x}_i) - \phi(\boldsymbol{x}_j)\right)^T \\ =& \boldsymbol{\Phi} \boldsymbol{L} \boldsymbol{\Phi}^T, \end{aligned} \qquad (4)$$

where $\boldsymbol{L} \in \mathbb{R}^{N \times N}$ is the graph Laplacian matrix defined by $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$, where $\boldsymbol{D} \in \mathbb{R}^{N \times N}$ is the (diagonal) degree matrix having elements $[\boldsymbol{D}]_{ii} = \sum_{i \neq j} [\boldsymbol{W}]_{ij}$, $i = 1, \ldots, N$, and $\boldsymbol{\Phi}$ is a matrix containing the data representations in $\mathcal{H}$. Depending on the exploited graph type [17], $\boldsymbol{L}$ 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) [17, 19, 20]. Finally, the Graph-Embedded SVM (GE-SVM) optimization problem is defined as follows [18, 20]:

$$\begin{aligned} \min_{\boldsymbol{w}, \boldsymbol{\xi}, b} \quad & \frac{1}{2} \|\boldsymbol{w}\|^2 + \frac{\lambda}{2} \boldsymbol{w}^T \boldsymbol{S} \boldsymbol{w} + c \sum_{i=1}^{N} \xi_i + b, \\ \text{s.t.} \quad & y_i \left(\boldsymbol{w}^T \phi(\boldsymbol{x}_i) + b\right) \leq 1 - \xi_i, i = 1, \ldots, N, \\ & \xi_i \geq 0, \end{aligned} \qquad (5)$$

while an additional constraint $\boldsymbol{w}^T \boldsymbol{S} \boldsymbol{w} > 0$ is also imposed demanding that the matrix $\boldsymbol{S}$ is positive semi-definite. Compared to standard SVM, an additional parameter $\lambda \geq 0$ is introduced, that controls the amount of regularization introduced by the second term. GE-SVM can be considered a generalization of other SVM-based methods, e.g., given a value of $\lambda = 0$, the method degenerates to standard SVM. Depending on the definition of $\boldsymbol{S}$, GE-SVM is equivalent to previously devised regularized SVM methods such as the Minimum Variance SVM [15] or Laplacian SVM [6].

The equivalent dual problem is defined as follows:

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^{N} \alpha_i - \\ & -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \phi(\boldsymbol{x}_i)^T \left(\boldsymbol{I} + \lambda \boldsymbol{S}\right)^{-1} \phi(\boldsymbol{x}_j), \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq c. \end{aligned} \qquad (6)$$

Finally, in order to classify a test sample, the standard SVM decision function is employed, by employing a regularized kernel of the following form:

$$\tilde{\kappa}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^T \left(\boldsymbol{I} + \lambda \boldsymbol{S}\right)^{-1} \phi(\boldsymbol{x}_j). \qquad (7)$$

GE-SVM can be solved using standard SVM implementations, by replacing the standard kernel matrix with the one defined above. As have been shown in recent work, GE-SVM outperforms the standard SVM [18, 20], in almost every SVM classification task, including one-class classification [19], and in some cases by a large extent. However, the increased classification performance comes with the cost of increased computational complexity, related to inefficient parameter tuning. The required parameters to

be tuned include the standard SVM parameter c and the introduced parameter λ, and moreover, depending on the adopted graph type, even more hyperparameters are required to be tuned as as well, e.g., k for the kNN graph case. Graph-hyperparameter selection is even more complex for the state-of-the-art performing positive and negative graph exploitation case [20]. The demanding computational complexity of GE-SVM limit its exploitation options in realistic application scenarios.

3. Multiplex data relationships in Support Vector Machines

In this Section, we describe in detail the proposed method, which extends the standard SVM problem, by incorporating additional optimization criteria, in addition to maximizing the classification margin. These criteria include incorporating geometric or semantic information about the training data, e.g., within-class variance information, local geometric data relationships information, expressed with multiple graph structures, i.e., multiplex data relationships. Their detailed mathematical description is given in Section 3.1. The introduced terms have the effect of projecting the SVM hyperplane in such directions, where the respective information of each additional term is emphasized. Moreover, a weighting parameter is introduced, that determines the contribution of each term to the final solution. From our derivations, analytically described in Subsection 3.2, it is proven that each of the proposed additional optimization term can also be expressed with a separate regularized kernel matrix. Thus, the proposed optimization problem can be solved using standard MKL-SVM methods, only by employing graph-regularized kernel matrices as base kernels, instead of standard ones, while the optimal weighting parameters are optimally estimated. Finally, computational complexity of the proposed method, as well as its generalization properties are discussed in Subsection 3.3.

3.1. Multiplex data relationships

Multiplex data relationships can be expressed by using a set of graphs, each describing a different pairwise property about the training set. Pairwise properties of the training data may include e.g., local geometric data information (encoded by kNN graphs) or global geometric data information (encoded in fully connected graphs). In addition, hand-crafted graph types or graphs that might be introduced in the future could be employed, as well. Let us denote by $\mathcal{G}^m = \{\mathcal{X}, \boldsymbol{W}^m\}, m = 1, \ldots, M$ the m-th graph with $\boldsymbol{W}^m$ its corresponding graph weight matrix, containing the weights of the connections between the graph vertices $\mathcal{X} = \{\phi(\boldsymbol{x}_1), \ldots, \phi(\boldsymbol{x}_N)\}$.

In order to express local geometric data information for our multiplex graph paradigm, let us denote by $\mathcal{G}^l$ a kNN graph. Also let $\mathcal{N}_i$ be the neighborhood of each vertex $\boldsymbol{x}_i$, connecting it with the k most similar vectors. Then, the corresponding graph weights can be initiated with a heat kernel function:

$$W_{ij}^l = \begin{cases} exp\left(-\gamma||\boldsymbol{x}_i - \boldsymbol{x}_j||_2^2\right), & \text{if } \boldsymbol{x}_j \in \mathcal{N}_i \\ 0, & \text{otherwise,} \end{cases} \quad (8)$$

where γ is a free parameter that scales the Euclidean distances between the graph vertices $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Let $\boldsymbol{S}_l$ encode the local geometry of the training data, defined in a similar manner as in (4):

$$\boldsymbol{S}_l = \boldsymbol{\Phi} \boldsymbol{L}_l \boldsymbol{\Phi}^T, \quad (9)$$

where $\boldsymbol{L}_l$ is the corresponding Laplacian matrix.

In order to encode the global geometry of the training data, fully connected graphs ($k = N$) of similar definition could be employed. Alternatively, we exploit a different fully connected graph type definition. From a disciminant analysis point of view [17], we would require that items belonging to the same class (e.g., class $c, c = 1, \ldots, C$) to be connected with equal weights, expressed in the graph $\mathcal{G}^w$, using the following weight matrix:

$$W_{ij}^w = 1/N_c, \text{ if } y_i = y_j, \quad (10)$$

where N_c is the number of items belonging to the $c-$th class. In fact, the corresponding matrix $\boldsymbol{S}_w$ that expresses global geometric data relationships as in equation (4), is the within-class scatter matrix, as

can be shown below:

$$\boldsymbol{S}_w = \sum_{c=1}^{C}\sum_{i=1}^{N_c}(\boldsymbol{\phi}_i^c - \bar{\boldsymbol{\phi}}^c)(\boldsymbol{\phi}_i^c - \bar{\boldsymbol{\phi}}^c)^T = \\ = \boldsymbol{\Phi}\left(\boldsymbol{I} - \sum_{c=1}^{C}\frac{1}{N_c}\boldsymbol{e}_c\boldsymbol{e}_c^T\right)\boldsymbol{\Phi}^T = \boldsymbol{\Phi}\boldsymbol{L}_w\boldsymbol{\Phi}^T, \quad (11)$$

where c is an index denoting the class of sample $\boldsymbol{x}_i$, $\boldsymbol{\phi}_i$ is a shorthand for $\phi(\boldsymbol{x}_i)$, $\bar{\boldsymbol{\phi}}^c$ is the mean sample of class c in the feature space, $\boldsymbol{e}_c \in \mathbb{R}^N$ is a vector of ones in the positions where $y_i = c$, or zeros, otherwise, and $\boldsymbol{L}_w$ is the corresponding graph Laplacian matrix.

In the following Subsection, we describe how multiplex data relationships are introduced to the SVM optimization problem.

3.2. Proposed method

The proposed method aims at generating a decision function in a space where multiplex data relationships are emphasized. In order to model the multiple data relationships, we employ the matrices $\boldsymbol{S}_m, m = 1, \ldots, M$, where the $m-$th matrix encode the data properties that are described by the $m-$th graph type. Then, a decision function can be obtained, by combining SVM hyperplanes $\boldsymbol{w}_m$ that have been regularized with the corresponding matrix $\boldsymbol{S}_m$, where the introduced regularization effect is controlled by the parameters $\lambda_m > 0$. Finally, multiplex data relationships are weighted according to their effect in the final decision function with the parameters μ_m. In order to determine the weighting parameters μ_m, and obtain the decision function at the same time, we propose the following optimization problem:

$$\begin{aligned}\min_{\{\boldsymbol{w}\},\boldsymbol{\xi},b,\boldsymbol{\mu}} \quad & \frac{1}{2}\sum_{m=1}^{M}\mu_m^{-p}\left(\|\boldsymbol{w}_m\|^2 + \lambda_m\boldsymbol{w}_m^T\boldsymbol{S}_m\boldsymbol{w}_m\right) + \\ & + c\sum_{i=1}^{N}\xi_i + b, \quad (12)\\ \text{s.t.} \quad & \sum_{m=1}^{M} y_i\left(\boldsymbol{w}_m^T\phi_m(\boldsymbol{x}_i) + b\right) \leq 1 - \xi_i, \\ & \xi_i \geq 0,\ \sum_{m=1}^{M}\mu_m^p = 1, \mu_m > 0,\end{aligned}$$

where each hyperplane $\boldsymbol{w}_m$, as well as each of the matrices $\boldsymbol{S}_m$ are defined in the feature space $\mathcal{H}_m$, and $p \geq 1$ is a parameter that affects the sparsity of the solution, similar to MKL methods. For simplicity reasons, we consider the case where $p = 1$, hereafter. The Lagrangian function corresponding to the proposed optimization problem is of the following form:

$$\begin{aligned}L = & \frac{1}{2}\sum_{m=1}^{M}\frac{1}{\mu_m}\boldsymbol{w}_m^T\left(\boldsymbol{I} + \lambda_m\boldsymbol{S}_m\right)\boldsymbol{w}_m + b - \\ & - \sum_{i=1}^{N}\alpha_i\left(\sum_{m=1}^{M} y_i\left(\boldsymbol{w}_m^T\phi_m(\boldsymbol{x}_i) + b\right) - 1 + \xi_i\right) + \\ & + \sum_{i=1}^{N}(c - \beta_i)\xi_i - \sum_{m=1}^{M}\gamma_m\mu_m - \delta\left(\sum_{m=1}^{M}\mu_m - 1\right),\end{aligned} \quad (13)$$

where $\alpha_i, \beta_i, \gamma_m$ and δ are the Lagrange multipliers corresponding to the constraints of (12) and $\boldsymbol{I}$ is an identity matrix of appropriate dimensions.

By setting the partial derivative of the Lagrangian with respect to each hyperplane equal to zero, $\frac{\partial L}{\partial \boldsymbol{w}_m} = 0$, we obtain:

$$\frac{1}{\mu_m}\left(\boldsymbol{I} + \lambda_m\boldsymbol{S}_m\right)\boldsymbol{w}_m = \sum_{i=1}^{N}\alpha_i y_i\phi(\boldsymbol{x}_i). \quad (14)$$

By setting the partial derivatives of L with respect to ξ_i and β equal to zero, i.e., $\frac{\partial L}{\partial \boldsymbol{\xi}} = 0$ and $\frac{\partial L}{\partial \boldsymbol{b}} = 0$, we obtain $\beta_i = c - \alpha_i$ and $\sum_{i=1}^{N}\alpha_i y_i = 1$, respectively. Then, by replacing back in the Lagrangian, the proposed optimization problem takes the following form:

$$\begin{aligned}\max_{\boldsymbol{\alpha}} \ \min_{\boldsymbol{\mu}} & \sum_{i=1}^{N}\alpha_i - \quad (15)\\ & -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j\left(\sum_{m=1}^{M}\mu_m\phi_m(\boldsymbol{x}_i)^T\left(\boldsymbol{I} + \lambda_m\boldsymbol{S}_m\right)^{-1}\phi_m(\boldsymbol{x}_j)\right) \\ & \text{s.t.} \quad 0 \leq \alpha_i \leq c \ \text{ and } \ \sum_{m=1}^{M}\mu_m = 1.\end{aligned}$$

We observe that the above defined optimization problem is similar to the standard SVM optimization problem, if we employ a kernel $q(\boldsymbol{x}_i, \boldsymbol{x}_j) =$

$\sum_{m=1}^{M} \mu_m \phi_m(\boldsymbol{x}_i)^T \left(\boldsymbol{I} + \lambda_m \boldsymbol{S}_m\right)^{-1} \phi_m(\boldsymbol{x}_j)$. This kernel can be explicitly determined by a linear combination of multiple base kernels $\tilde{\kappa}_m$, weighted by parameters μ_m, such that:

$$q(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{m=1}^{M} \mu_m \tilde{\kappa}_m(\boldsymbol{x}_i, \boldsymbol{x}_j), \tag{16}$$

where $\tilde{\kappa}_m(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi_m(\boldsymbol{x}_i)^T \left(\boldsymbol{I} + \lambda_m \boldsymbol{S}_m\right)^{-1} \phi_m(\boldsymbol{x}_j)$ contains data similarity in the space where the $m-$th training data property is emphasized. We recall that $\boldsymbol{S}_m = \boldsymbol{\Phi} \boldsymbol{L}_m \boldsymbol{\Phi}^T$, where $\boldsymbol{L}_m$ is the Laplacian matrix of the $m-$th graph. In order to obtain the base kernel matrix, we first calculate the inversion $\left(\boldsymbol{I} + \lambda_m \boldsymbol{S}_m\right)^{-1}$, by exploiting the Woodbury matrix inversion identity [31]:

$$\left(\boldsymbol{I} + \lambda_m \boldsymbol{\Phi} \boldsymbol{L}_m \boldsymbol{\Phi}^T\right)^{-1} = \boldsymbol{I} - \boldsymbol{\Phi} \left(\frac{1}{\lambda_m} \boldsymbol{L}_m^{-1} + \boldsymbol{\Phi}^T \boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^T, \tag{17}$$

where $\boldsymbol{\Phi}^T \boldsymbol{\Phi} = \boldsymbol{K}$, which is a Kernel matrix that expresses similarity in the space associated with the employed mapping function. Moreover, this formula can be further simplified by exploiting the Searle matrix inversion identity [31]:

$$\left(\frac{1}{\lambda_m} \boldsymbol{L}_m^{-1} + \boldsymbol{K}\right)^{-1} = \boldsymbol{K}^{-1} \left(\lambda_m \boldsymbol{L}_m + \boldsymbol{K}^{-1}\right)^{-1} \lambda_m \boldsymbol{L}_m, \tag{18}$$

Finally, each regularized kernel matrix can be explicitly calculated as follows:

$$\begin{aligned} \tilde{\boldsymbol{K}}_m &= \boldsymbol{\Phi}^T \left[\boldsymbol{I} - \boldsymbol{\Phi} \boldsymbol{K}^{-1} \left(\lambda_m \boldsymbol{L}_m + \boldsymbol{K}^{-1}\right)^{-1} \lambda_m \boldsymbol{L} \boldsymbol{\Phi}^T\right] \boldsymbol{\Phi} = \\ &= \boldsymbol{K} - \left(\lambda_m \boldsymbol{L}_m + \boldsymbol{K}^{-1}\right)^{-1} \lambda_m \boldsymbol{L}_m \boldsymbol{K} = \\ &= \left[\boldsymbol{I} - \left(\lambda_m \boldsymbol{L}_m + \boldsymbol{K}^{-1}\right)^{-1} \lambda_m \boldsymbol{L}_m\right] \boldsymbol{K}. \end{aligned} \tag{19}$$

By replacing the calculated base kernels back to the Lagrangian, we obtain a MKL-SVM optimization problem:

$$\max_{\boldsymbol{\alpha}} \min_{\boldsymbol{\mu}} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \sum_{m=1}^{M} \mu_m \tilde{\kappa}_m(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{20}$$

$$\text{s.t.} \quad 0 \leq \alpha_i \leq c \quad \text{and} \quad \sum_{m=1}^{M} \mu_m = 1,$$

which is similar to the optimization problem defined in (2), only by replacing the base kernels $\boldsymbol{K}_m$ with $\tilde{\boldsymbol{K}}_m$. In order to solve this optimization problem, any MKL-SVM method can be employed [12]. To this end, we have employed the recently proposed soft-margin MKL-SVM method [11] in all our experiments, since it outperforms other widely adopted MKL methods [32, 33] in video classification problems, by providing an efficient compromise between sparse solutions and fast convergence. That is, the min-max optimization problem is broken into two quadratic programming optimization problems that are solved sequentially, one for the standard SVM, and a separate soft-margin optimization one for determining the parameters μ_m. Finally, in order to classify a test sample, we employ the MKL decision function (3), using the appropriate matrices.

3.3. Discussion

The proposed method employs multiple graphs for regularization purposes, in the form of multiple single-graph regularized kernels. The optimization problem is formulated as a MKL-SVM optimization problem. The advantage of our approach is the elimination of exhaustive parameter fine-tuning, related to graph-hyper-parameters. Their effect, along with the parameter λ, are implicitly determined only by optimally calculating the kernel contribution parameters μ_m, inside a separate optimization problem. In order to demonstrate how important is this property, let us consider the following example. Let a set of M kNN graphs with weights initiated with an RBF heat kernel function, that are available to be exploit in the SVM optimization process. The parameters for each graph include the number of nearest neighbors k and the RBF parameter γ_k. By introducing them in the SVM problem, we have another additional RBF γ parameter for the SVM kernel function, the amount of the introduced regularization λ and the standard SVM parameter c, totaling 5 parameters. Without an optimization procedure, i.e., the proposed approach, determining the optimal parameter combination with traditional methods, e.g., grid search, is computationally intensive.

On the other hand, we consider the complexity of the proposed method. Since the proposed method

can be solved using any MKL-SVM solver, its computational complexity in the training phase is equal to the complexity of the solver, along with the complexity required to calculate the regularized basekernels using equation (19). Let us consider that all of the employed kernels are regularized versions of the same standard basekernel $\boldsymbol{K}$ (e.g., RBF), having size equal to $N \times N$, where N is the number of the employed training data. First, the basekernels need to calculated and inverted. Then, in order to obtain each regularized basekernel version, a Laplacian matrix $\boldsymbol{L}_m$ of size $N \times N$ needs to be determined. Then, an additional inversion of the quantity inside the parenthesis of size $N \times N$ is required. Finally, this quantity is multiplied with $\boldsymbol{L}_m$, and this result is then saved and stored, since this result will be employed for deriving the regularized basekernel at the inference stage, as well. Therefore, the complexity of the training stage is equal to the complexity of the MKL-SVM solver, plus two inversions of size $N \times N$, the calculation of the Laplacian matrix and two matrix multiplications of size $N \times N$ for each basekernel. In the inference stage, the computational complexity is equal to standard MKL-SVM, plus one matrix multiplication for each of the resulted basekernels. Thus, assuming hardware restrictions e.g., embedded systems, adopting sparse solutions is preferred.

Finally, another aspect of the proposed method includes its generalization features. The proposed formulation is general, since related methods may be implemented as special cases of the proposed method, hereafter. That can be achieved by changing the basekernel matrix combination. For example, by replacing the derived kernel matrix $\boldsymbol{Q}$ with the standard SVM kernel matrix $\boldsymbol{K}$ and $\mu = 1$, the proposed method degenerates to standard SVM. By using a set of standard SVM kernel matrices derived by employing several mapping functions, or similar mapping functions with different parameters, the proposed method represents the basic MKL formulation. Finally, by introducing only a single graph ($\mu = 1$) in the SVM optimization process, the proposed method degenerates to GE-SVM.

4. Experiments

In order to evaluate the performance of the proposed method, we have conducted experiments in visual analysis classification problems. To this end, we have employed publicly available datasets for face recognition, object classification, human action recognition. The employed datasets were carefully selected to demonstrate the effectiveness of the proposed method in various circumstances, i.e., various type of input is given to the proposed method, including pre-extracted feature vectors, deep features, pre-computed kernel matrices, features having minimal pre-processing i.e., pixel luminosities and handcrafted features. Since all employed datasets are well balanced in terms of instances per class, for both training and testing purposes, the Classification Rate (CR) was employed as performance metric.

Along with the proposed method, we have also implemented the standard SVM [5], the GE-SVM [18] and MKL-SVM [11]. For comparison fairness, the same SVM solver was employed for all methods [34], and the parameter settings were also set to be equal for all methods, where applicable. Our experimental platform was a PC with 32GB of RAM on a i7 processor, using a Matlab implementation. In all our experiments, we have employed the kernel versions of the competing algorithms for each experiment, by employing the RBF kernel:

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|_2^2\right), \qquad (21)$$

where $\gamma = 1/2a\sigma^2$, σ^2 is the standard deviation of the training data, which is the normal scaling factor and the optimal γ was determined by setting values to a equal to $a = -1, 0, 0.5, 1, 5, 10$. The SVM parameter c was set equal to 10^{ℓ} , $\ell = -2, \ldots, 6$. The optimal parameter settings for each method were determined using a $5-$fold cross validation procedure on the training set. In MKL-SVM, all RBF kernel matrices were employed in the training phase. Regarding GE-SVM, the additional parameter λ was set equal to $10^s, s = -3, \ldots, 3$, and two types of regularizers were employed, i.e., $\boldsymbol{S}_l$ from equation (9) and $\boldsymbol{S}_w$ from equation (11). The kNN graph being employed in $\boldsymbol{S}_l$ was containing local geometric data

relationships from $k = 5, 10, 15$ neighbors. In GE-SVM, the best performing regularized kernel during cross validation was employed for testing the classifier. All of the constructed regularized kernels constructed for GE-SVM were also employed in the proposed method, with the difference that only a value $\lambda_m = 10^{-1}$ was used, since its effects are controlled by the parameters μ_m.

Detailed description for the experimental protocol followed for each classification problem is analytically described in Sections 4.1, 4.2 and 4.3, respectively. Finally, we describe the conducted significance analysis of the obtained results in Section 4.4.

4.1. Experiments in face recognition

In our experiments in face recognition, we have employed the PubFig+LFW [35], AR [36], Yale [37] and ORL [38] datasets. The PubFig+LFW [35] is a benchmark dataset for open-universe face identification, consisting of $13,002$ 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 form the distractor set from LFW. For each facial image, the extracted features include the Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP) and Gabor wavelet features. The extracted features were reduced to 2048 dimensions with Principal Component Analysis (PCA), from which we only employed the first 1536 dimensions, as adviced by the dataset providers [35]).

Moreover, we have also employed classic face recognition datasets, i.e. the AR [36], Yale [37] and ORL [38] datasets, containing 2600, and 2432 and 400 frontal facial images belonging to 100, 38 and 40 subjects, respectively. As feature vectors, we have employed the grayscale resized images to 40×30 pixels, and vectorized them so that to produce a $D = 1200$ dimensional vector for each facial image. Since no standard experimental protocol have been defined on these datasets, we have performed a $5-$fold cross-validation procedure and report the average obtained performance among the folds.

Experimental results are drawn in Table 1. As can be seen, the proposed method outperformed all competing methods in every case, in terms of classification accuracy. More specifically, by observing the performance of all competing methods in PubFig+LFW dataset, whose feature vectors include information from hand-crafted descriptors, employing Multiple Kernel matrices seem to have been beneficial to classification performance. This can be explained by the fact that the extracted features may lie in multiple distributions, not modeled adequately by a single normal distribution (i.e., the standard SVM case), or even a regularized one (i.e., the GE-SVM case). The performance of MKL-SVM denoted that exploiting multiple distributions for modeling data similarity was beneficial to performance. In every case, the proposed method outperformed the competition, by exploiting the additional global and local geometric particularities of each class, modeled by the added graph structures. This information acted as an advanced regularizer to the solution, offering more accurate feature representation, in comparison with the competition.

In our experiments on classic face recognition datasets, we have observed that employing the MKL-SVM, seem to have not influenced positively the classification performance, maybe related to over-fitting issues. This effect is supported by the performance of GE-SVM, which outperformed the standard SVM and MKL-SVM, by having one graph regularizing the classification space. However, the proposed method was able to alleviate the negative over-fitting effects, by optimally determining the most efficient regularized kernel combination.

Table 1: Classification rates (CR) in Face Recognition datasets

Algorithm/Dataset	PubFig+LFW	ORL	AR	Yale
SVM	36.24	98.75	99.11	97.94
GE-SVM	34.35	98.75	99.19	97.94
MKL-SVM	84.17	98.75	90.57	96.08
PROPOSED	**88.77**	**99.25**	**99.42**	**98.06**

4.2. Experiments in object classification

In our experiments on object classification, we have employed the CIFAR-100 [39] and Caltech101 [40] datasets.

In CIFAR-100 dataset, we have employed pre-extracted features [41]. That is, the feature vectors were computed by performing a forward pass to a pre-trained CNN network from the fully connected layer 'fc2', having feature dimensionality $D = 255$, based on a Hadamard coding preprocessing [41, 42]. We have employed the small dataset version, which includes 5000 training and 1000 testing samples, belonging to 10 classes, corresponding to ones predefined by the dataset providers [39]. We have constructed the RBF kernel matrices by employing the above mentioned features, and employed them to the SVM and MKL-SVM methods. Their regularized alternatives were employed for the GE-SVM and the proposed method. In Caltech101 dataset, we have employed 10 pre-computed kernel matrices [43], derived from employing the Geometric blur [44], dense visual words [45] and Self-similarity [46] descriptors.

In standard SVM, we have reported the maximum performance obtained by employing each of the 10 pre-computed kernel matrices. In GE-SVM, we have employed the regularized kernel versions, by employing $\boldsymbol{S}_l$ with $k = 5, 10, 15$ neighbors and $\boldsymbol{S}_w$ with $\lambda_m = 10^{-1}$, leading to a total of 40 kernel matrices. Finally, these kernel matrices were employed by proposed method, as well. Classification rates on both datasets is shown in Table 2. As can be seen in both cases, the proposed method greatly outperformed the competition. The proposed method outperformed MKL-SVM by 1.5%. This demonstrates the effectiveness of the proposed method, for the case where pre-computed kernel matrices have been employed.

Table 2: Classification rates (CR) in object recognition datasets

Algorithm/Dataset	CIFAR-100	Caltech101
SVM	73.20	66.17
GE-SVM	72.30	66.56
MKL-SVM	75.40	72.42
PROPOSED	**79.80**	**73.39**

4.3. Experiments in human action recognition

In our experiments in human action recognition, we have employed the the i3DPost multi-view action database [47], the IMPART Multi-modal/Multi-view Dataset [48], the Olympic Sports [49] and the Hollywood3D [50] publicly available datasets.

In i3DPost and IMPART datasets, we have employed a 3-fold cross validation procedure, where we have split the datasets in 3 mutually exclusive sets, having 6/8 people for training purposes, and 2/8 for testing in i3DPost dataset, and 2/3 and 1/3 in IMPART dataset, respectively. This procedure was repeated 3 times, and the reported performance is the average obtained classification rate among the 3 folds. In Hollywood 3D and Olympic Sports datasets, we employed the standard training and test videos, provided by the dataset providers [49, 50, 51]. In order to obtain vectorial video representations for each video segment depicting each action, we have employed the dense trajectory-based video description [52]. This video description calculates five descriptor types, namely the Histogram of Oriented Gradients, Histogram of Optical Flow, Motion Boundary Histogram along direction x, Motion Boundary Histogram along direction y and the normalized trajectory coordinates 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). Thereby, each video segment is then described by 5 vectors. We have employed these video segment descriptors in order to obtain five video segment representations by using the Bag-of-Words model [53, 54], creating a video description of having 5 descriptors of $100, 500, 4000$ and 4000 dimensions, for i3DPost, IMPART, Olympic sports and Hollywood3D, respectively. In GE-SVM and standard SVM methods, in order to fuse infromation from all descriptors, we combined the 5 descriptor types with kernel methods using a late fusion approach [55], i.e.,:

$$k(\mathcal{X}_i, \mathcal{X}_j) = exp\left(-\frac{1}{d}\gamma_d \sum_d \|\mathbf{x}_i^d - \mathbf{x}_j^d\|_2^2\right), \quad (22)$$

$\boldsymbol{x}_i^d \in \mathbb{R}^D$ is a video feature vector for $d = 5$ (number of descriptor types) and $\gamma_d = 2\sigma_d^2$ is a parameter scaling the Euclidean distance between $\boldsymbol{x}_i^d$ and $\boldsymbol{x}_j^d$. In MKL-SVM and the proposed method, besides the fused kernel matrix, each separate kernel matrix con-

taining data similarity from each descriptor type was also given as input.

Experimental results in human action recognition are drawn in Table 3. As can be observed, the proposed method outperformed the competition in almost every case. Using MKL-SVM for fusing information from the specific descriptor types provided slightly improved classification performance in comparison with standard SVM. In addition, by observing the performance of GE-SVM, employing graph-based regularization provided furthermore increased classification performance. The proposed method combined the performance gains by both worlds, consisting itself superior from the competition.

Table 3: Classification rates (CR) in Human Action Recognition datasets

Algorithm/Dataset	I3DPost	IMPART	Olympic Sports	Hollywood 3D
SVM	94.39	85.32	73.13	29.87
GE-SVM	94.87	**86.47**	**74.63**	29.87
MKL-SVM	94.39	85.33	73.88	30.52
PROPOSED	**95.51**	85.75	**74.63**	**32.14**

4.4. Significance analysis

After obtaining the performance of the competing methods in all experiments, we determined whether the observed differences of the proposed method with the competition are statistically significant, or not, by using the implementation from [56, 57, 58]. To this end, we have tested the null hypotheses that all classifiers perform the same, using the Friedman's test. The mean ranks for each algorithm according to their performance in all classification problems are shown in Table 4. By employing 10 datasets and 4 classifiers, the degrees of freedom is equal to 27. The Friedman statistic is equal to $\chi^2_{\mathcal{F}} = 16.14$, and the critical value was 7.81. Therefore, the null hypotheses that all classifiers perform the same, was rejected. After employing the Nemenyi post-hoc procedure for pairwise comparison, using a significance level of 95%, i.e., $a = 0.05$, the Critical Distance (CD) was found at 1.48, which means that the proposed method performed significantly better than all competing methods. Moreover, we have also used the Bergman-Hommel's posthoc procedure, which amplifies the test power by using an exhaustive sets of hypothesis, i.e hypothesis that can be true at the same time. The critical distance was calculated at 1.38. Therefore, the proposed method performs significantly better than the competition.

Table 4: Statistical test details

Mean ranks	SVM	GE-SVM	MKL-SVM	Proposed
	3.35	2.65	2.85	1.15
Posthoc Procedure	Nemenyi		Bergman-Hommel's	
α	0.05		0.0083	
CD	1.48		1.38	

5. Conclusion

We have presented a novel method for introducing multiplex data relationships to the SVM optimization process, by exploiting pairwise data information expressed in multiple graph structures. Our experiments denoted that the proposed method provided increased classification performance consistently against related methods, in different visual data classification problems. The improved classification accuracy was mainly achieved, due to the exploitation of advanced graph-based regularization settings in an optimal fashion, representing the multimodal/multiplex image and video data characteristics. Since the proposed method provided enhanced classification performance using different descriptor settings, it may perform well in other standard classification problems, as well.

Moreover, since the proposed method is a generic formulation for Graph-based SVM methods and Multiple Kernel methods, evolution in both fields shall favor the proposed method as well. That is, by including advanced regularization settings using novel graph combinations or improved Multiple Kernel Learning solvers. This can also serve as a feature research direction.

Acknowledgment

This work has received funding from the European Union's European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement

number 316564 (IMPART) and Horizon 2020 research and innovation programme under grant agreement No 731667 (MULTIDRONE). This publication reflects only the authors' views. The European Commission is not responsible for any use that may be made of the information it contains.

This is the author pre-print version. Please cite the publisher maintained version in your work. DOI link: **https://doi.org/10.1016/J.PATCOG.2018.07.032**
Copyright: 2018 Elsevier Ltd.

References

[1] H. Yang, L. Shao, F. Zheng, L. Wang, Z. Song, Recent advances and trends in visual tracking: A review, Neurocomputing 74 (18) (2011) 3823–3831.

[2] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M.-M. Cheng, S. L. Hicks, P. H. Torr, Struck: Structured output tracking with kernels, IEEE transactions on pattern analysis and machine intelligence 38 (10) (2016) 2096–2109.

[3] C.-W. Hsu, C.-J. Lin, A comparison of methods for multiclass support vector machines, IEEE transactions on Neural Networks 13 (2) (2002) 415–425.

[4] B. Schölkopf, 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.

[5] A. J. Smola, B. Schölkopf, A tutorial on support vector regression, Statistics and computing 14 (3) (2004) 199–222.

[6] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: A geometric framework for learning from labeled and unlabeled examples, Journal of machine learning research 7 (Nov) (2006) 2399–2434.

[7] B. Schölkopf, R. Herbrich, A. J. Smola, A generalized representer theorem, International Conference on Computational Learning Theory (2001) 416–426.

[8] A. Rakotomamonjy, F. R. Bach, S. Canu, Y. Grandvalet, Simplemkl, Journal of Machine Learning Research 9 (Nov) (2008) 2491–2521.

[9] M. Kloft, U. Brefeld, S. Sonnenburg, A. Zien, Lp-norm multiple kernel learning, Journal of Machine Learning Research 12 (Mar) (2011) 953–997.

[10] C. Cortes, M. Mohri, A. Rostamizadeh, Learning non-linear combinations of kernels, Advances in neural information processing systems (2009) 396–404.

[11] X. Xu, I. W. Tsang, D. Xu, Soft margin multiple kernel learning, IEEE transactions on neural networks and learning systems 24 (5) (2013) 749–761.

[12] M. Gönen, E. Alpaydın, Multiple kernel learning algorithms, Journal of Machine Learning Research 12 (Jul) (2011) 2211–2268.

[13] S. S. Bucak, R. Jin, A. K. Jain, Multiple kernel learning for visual object recognition: A review, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (7) (2014) 1354–1369.

[14] 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.

[15] S. Zafeiriou, A. Tefas, I. Pitas, Minimum class variance support vector machines, IEEE Transactions on Image Processing 16 (10) (2007) 2551–2564.

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

[17] 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.

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

[19] V. Mygdalis, A. Iosifidis, A. Tefas, I. Pitas, Graph embedded one-class classifiers for media data classification, Pattern Recognition 60 (2016) 585 – 595. doi:http://dx.doi.org/10.1016/j.patcog.2016.05.033.

[20] A. Iosifidis, M. Gabbouj, Multi-class support vector machine classifiers using intrinsic and penalty graphs, Pattern Recognition 55 (2016) 231–246.

[21] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, M. A. Porter, Multilayer networks, Journal of complex networks 2 (3) (2014) 203–271.

[22] M. Wang, X.-S. Hua, R. Hong, J. Tang, G.-J. Qi, Y. Song, Unified video annotation via multigraph learning, IEEE Transactions on Circuits and Systems for Video Technology 19 (5) (2009) 733–746.

[23] J. J.-Y. Wang, H. Bensmail, X. Gao, Multiple graph regularized nonnegative matrix factorization, Pattern Recognition 46 (10) (2013) 2840–2847.

[24] J. Zhou, Y. Ren, Y. Yan, L. Pan, A multiple graph label propagation integration framework for salient object detection, Neural Processing Letters (2015) 1–19.

[25] P. Chen, L. Jiao, F. Liu, J. Zhao, Z. Zhao, S. Liu, Semi-supervised double sparse graphs based discriminant analysis for dimensionality reduction, Pattern Recognition 61 (2017) 361–378.

[26] B. Fan, Y. Cong, Consistent multi-layer subtask tracker via hyper-graph regularization, Pattern Recognition 67 (2017) 299–312.

[27] C. Jose, P. Goyal, P. Aggrwal, M. Varma, Local deep kernel learning for efficient non-linear svm prediction, Proceedings of the 30th international conference on machine learning (ICML) (2013) 486–494.

[28] H. Xia, S. C. Hoi, R. Jin, P. Zhao, Online multiple kernel similarity learning for visual search, IEEE transactions on pattern analysis and machine intelligence 36 (3) (2014) 536–549.

[29] M. GöNen, E. AlpaydıN, Localized algorithms for multiple kernel learning, Pattern Recognition 46 (3) (2013) 795–807.

[30] A. Shrivastava, V. M. Patel, R. Chellappa, Multiple kernel learning for sparse representation-based classification, IEEE Transactions on Image Processing 23 (7) (2014) 3013–3024.

[31] K. B. Petersen, M. S. Pedersen, et al., The matrix cookbook, Technical University of Denmark 7 (15) (2008) 510.

[32] M. Kloft, U. Brefeld, P. Laskov, K.-R. Müller, A. Zien, S. Sonnenburg, Efficient and accurate lp-norm multiple kernel learning, Advances in neural information processing systems (2009) 997–1005.

[33] M. Varma, B. R. Babu, More generality in efficient multiple kernel learning, Proceedings of the 26th Annual International Conference on Machine Learning (2009) 1065–1072.

[34] C.-C. Chang, C.-J. Lin, LIBSVM: A library for support vector machines, ACM Transactions on Intelligent Systems and Technology 2 (2011) 27:1–27:27, software available at `http://www.csie.ntu.edu.tw/ cjlin/libsvm`.

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

[36] A. M. Martinez, The ar face database, In: CVC Technical Report 24.

[37] 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.

[38] F. Samaria, A. Harter, Parameterisation of a stochastic model for human face identification, In: IEEE Workshop on Applications of Computer Vision (1994).

[39] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images.

[40] L. Fei-Fei, R. Fergus, P. Perona, Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories, Computer vision and Image understanding 106 (1) (2007) 59–70.

[41] S. Yang, P. Luo, C. C. Loy, K. W. Shum, X. Tang, et al., Deep representation learning with target coding., In: AAAI Conference on Artificial Intelligence (2015) 3848–3854.

[42] N. Srivastava, R. R. Salakhutdinov, Discriminative transfer learning with tree-based priors, Advances in Neural Information Processing Systems (2013) 2094–2102.

[43] A. Vedaldi, V. Gulshan, M. Varma, A. Zisserman, Multiple kernels for object detection, In: International Conference on Computer Vision (2009) 606–613.

[44] A. C. Berg, T. L. Berg, J. Malik, Shape matching and object recognition using low distortion correspondences, In: Computer Vision and Pattern Recognition (CVPR) 1 (2005) 26–33.

[45] A. Bosch, A. Zisserman, X. Munoz, Representing shape with a spatial pyramid kernel, In: ACM international conference on Image and video retrieval (2007) 401–408.

[46] E. Shechtman, M. Irani, Matching local self-similarities across images and videos, In: Computer vision and pattern recognition (CVPR) (2007) 1–8.

[47] N. Gkalelis, H. Kim, A. Hilton, N. Nikolaidis, I. Pitas, The i3DPost multi-view and 3D human action/interaction database, In: European Conference on Visual Media Production (CVMP) (2009) 159–168.

[48] H. Kim, A. Hilton, Influence of colour and feature geometry on multimodal 3D point clouds data registration, In: International Conference on 3D Vision (3DV) (2015) 202–209.

[49] J. C. Niebles, C.-W. Chen, L. Fei-Fei, Modeling temporal structure of decomposable motion segments for activity classification, In: European conference on computer vision (2010) 392–405.

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

[51] M. Marszalek, I. Laptev, C. Schmid, Actions in context, In: Computer Vision and Pattern Recognition (CVPR) (2009) 2929–2936.

[52] 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.

[53] A. Iosifidis, A. Tefas, I. Pitas, Discriminant bag of words based representation for human action recognition, Pattern Recognition Letters 49 (2014) 185–192.

[54] N. Passalis, A. Tefas, Information clustering using manifold-based optimization of the bag-of-features representation, IEEE transactions on cybernetics.

[55] 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.

[56] A. Ulaş, O. T. Yıldız, E. Alpaydın, Cost-conscious comparison of supervised learning

algorithms over multiple data sets, Pattern Recognition 45 (4) (2012) 1772–1781. doi:http://dx.doi.org/10.1016/j.patcog.2011.10.005.

[57] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine learning research 7 (Jan) (2006) 1–30.

[58] S. Garcia, F. Herrera, An extension on"statistical comparisons of classifiers over multiple data sets"for all pairwise comparisons, Journal of Machine Learning Research 9 (Dec) (2008) 2677–2694.