Multi-class Support Vector Machine Classifiers using Intrinsic and Penalty Graphs

Alexandros Iosifidis and Moncef Gabbouj

Department of Signal Processing, Tampere University of Technology,
P. O. Box 553, FIN-33720 Tampere, Finland
e-mail: {alexandros.iosifidis,moncef.gabbouj}@tut.fi

Abstract

In this paper, a new multi-class classification framework incorporating geometric data relationships described in both intrinsic and penalty graphs in multi-class Support Vector Machine is proposed. Direct solutions are derived for the proposed optimization problem in both the input and arbitrary-dimensional Hilbert spaces for linear and non-linear multi-class classification, respectively. In addition, it is shown that the proposed approach constitutes a general framework for SVM-based multi-class classification exploiting geometric data relationships, which includes several SVM-based classification schemes as special cases. The power of the proposed approach is demonstrated in the problem of human action recognition in unconstrained environments, as well as in facial image and standard classification problems. Experiments indicate that by exploiting geometric data relationships described in both intrinsic and penalty graphs the SVM classification performance can be enhanced.

Keywords:  Multi-class Classification, Maximum Margin Classification, Support Vector Machine, Graph Embedding.
1. Introduction

Support Vector Machine (SVM) [43] is a standard classification technique that has been shown to provide state-of-the-art performance in many classification problems. In addition to their good generalization ability, the popularity of SVMs is a consequence of their ability to represent the classification problem at hand as a quadratic convex optimization problem, leading to a global optimal solution, while non-linear decision functions can be learned by exploiting the well-known kernel trick [31, 39, 25, 36].

The standard SVM is a binary classifier that learns a hyperplane separating two classes with maximum margin. While the maximum margin property of the SVM classifier is very powerful, it has been shown that enhanced performance can be achieved by incorporating geometric data information in the SVM optimization process. This is due to the fact that, by exploiting such additional information, the classifier takes into account geometric properties of the classes in addition to the position of the support vectors. Specifically, it has been shown that the incorporation of the intra-class variance information (described by the corresponding within-class scatter matrix) in the SVM optimization problem leads to enhanced performance in frontal face verification [42], as well as in various other classification problems, e.g. gender determination, eye glass detection and neutral facial expression recognition and standard classification problems [50, 45, 51, 19]. In addition, it has been shown that the exploitation of intrinsic graph structures defined under the Graph Embedding framework [49] further enhances the perfor-
mance of the resulting classifier [1]. Graph Embedding, is a general framework which can be exploited in order to define Subspace Learning techniques, such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Marginal Discriminant Analysis (MDA) and Local Fisher Discriminant Analysis (LFDA). This is achieved by defining an intrinsic graph expressing properties of the data that are subject to minimization (e.g. the within-class variance in the case of LDA) and a penalty graph expressing properties of the data that are subject to maximization (i.e. the between-class variance in the case of LDA).

In all the above mentioned SVM-based classification approaches, the One-Versus-Rest (OVR) or One-Versus-One (OVO) binary classifier combination schemes are employed in order to perform multi-class classification [35]. That is, for a classification problem formed by data belonging to $K$ classes, multiple\(^1\) binary classifiers are trained, each of which solves a sub-problem of the original multi-class classification problem. In the test phase, a test sample is introduced to all the binary classifiers and their responses are combined in order to provide the final classification result [30]. Such an approach inherently sets the assumption that the classification problems solved by the various binary classifiers are independent. In order to overcome this assumption, multi-class SVM formulations and their counterparts incorporating the within-class variance of the training data in a multi-class SVM formulation, have been proposed in [47, 48, 21, 22, 3].

\(^1\)The number of binary classifiers is equal to $K$ for the OVR and $\frac{K(K-1)}{2}$ for the OVO combination schemes. For the standard SVM formulation, combined OVR and OVO classification schemes have also been used [20, 27], where a model formed by $K \leq M \leq \frac{K(K-1)}{2}$ binary classifiers is created.
In [21, 22], only the case where the within-class variance of the training data is exploited for SVM-based multi-class classification is proposed. In addition, the non-linear extension of the proposed classifiers is achieved by applying a two-step process. Specifically, in [21], the training data are projected to the range of the within-class scatter matrix first and standard multi-class SVM classification is applied on the projected data [47, 48]. In [22], it is shown that the kernel formulation of the proposed multi-class classifier is equivalent to applying kernel PCA on the training data, followed by the application of the proposed linear classifier exploiting the within-class variance of the projected training data.

In this paper, we propose a new optimization problem for SVM-based multi-class classification exploiting geometric information of the training data. To do this, we incorporate geometric data information described in both intrinsic and penalty graphs as designed in the context of the Graph Embedding framework. Compared to the solution proposed in [1] exploiting only intrinsic graphs for binary classification problems, the proposed classifier exploits general graph structures expressing both intrinsic (expressing data relationships to be minimized) and penalty (expressing data relationships to be maximized) criteria, under a multi-class SVM formulation. Compared to the solutions proposed in [21, 22] exploiting only the within-class variance of the training data for multi-class classification, the proposed approach is able to exploit more generic intrinsic graph structures, as well as penalty ones. In addition, we propose a direct solution for the optimization problem solved for non-linear data classification. Finally we show that the proposed approach constitutes a general framework for SVM-based multi-class
classification exploiting geometric information of the training data and that the methods in [42, 50, 1, 35, 21, 22, 45] are special cases of the proposed approach.

We apply the proposed method in facial image and standard classification problems and to the problem of human action recognition in unconstrained environments, usually also referred to as ‘action recognition in the wild’. A lot of research has been conducted in this area during the last decade. The interested reader may refer to [26]. Perhaps the most well studied and successful approach for action representation is based on the Bag of Words (BoWs) model [11]. According to this model, each action video is represented by a vector obtained by applying quantization on the features describing it and using a set of feature prototypes forming the so-called codebook. This codebook is usually determined by clustering the features describing training action videos, while discriminative codebook construction methods have also been recently proposed [12]. This approach has been tested in most of the existing benchmark datasets and its efficacy has been proven, since it provides state-of-the-art performance in most cases. We follow the state-of-the-art approach [46] describing videos depicting actions by using five descriptor types, i.e. Histogram of Oriented Gradient (HOG), Histogram of Optical Flow (HOF), Motion Boundary Histogram along the direction x (MBHx), Motion Boundary Histogram along the direction y (MBHy) and (normalized) Trajectory, evaluated on the trajectories of densely sampled interest points. Such an action description has been evaluated in most of the existing benchmark datasets, where it has been shown that it provides satisfactory performance (state-of-the-art in most cases).
In summary, the contributions of the paper are as follows:

- A new optimization problem for SVM-based multi-class classification is proposed that exploits geometric data relationships described in both intrinsic and penalty graphs.

- A new direct solution is proposed for the optimization problem used to determine non-linear decision functions for multi-class classification.

- The proposed approach is shown to constitute a general framework for SVM-based classification exploiting geometric data information that includes several SVM-based classifiers as special cases.

The reminder of the paper is organized as follows. We provide an overview of related previous work in Section 2. The proposed method is described in detail in Section 3. Experiments conducted in order to evaluate its performance are described in Section 4. Conclusions are drawn in Section 5.

2. Previous Work

Let us denote by \( \{x_i, l_i\}, \ i = 1, \ldots, N \) a set of \( D \)-dimensional vectors \( x_i \) and the corresponding class labels \( l_i \in \{1, \ldots, K\} \). We would like to train a multi-class classification scheme that is able to classify a test vector \( x_t \in \mathbb{R}^D \) to one of the \( K \) classes.

2.1. Binary SVM classifier

As previously described, multi-class classification can be achieved by training multiple binary classifiers [30]. Let us define the binary labels \( y_i \in \{-1, 1\} \) deter-
mining whether the vectors \( x_i \) belong to the positive or negative class of the binary classification problem at hand. In SVM, the optimal separating hyperplane is the one that separates the two classes with maximum margin. The SVM optimization problem is defined as:

\[
\min_{w, b} \frac{1}{2} w^T w + c \sum_{i=1}^{N} \xi_i,
\]

subject to the constraints:

\[
y_i (w^T x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, N;
\]

where \( w \in \mathbb{R}^D \) is the vector defining the separating hyperplane, \( b \) represents the offset of the hyperplane from the origin, \( \xi_i, i = 1, \ldots, N \) are the so-called slack variables and \( c > 0 \) is a regularization parameter denoting the importance of the training error in the optimization problem. The solution of the above-described optimization problem is a quadratic convex optimization problem of the form:

\[
\max_{\alpha} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j,
\]

subject to the constraint \( 0 \leq \alpha_i \leq c, i = 1, \ldots, N \). \( \alpha \in \mathbb{R}^N \) is a vector containing the Lagrange multipliers \( \alpha_i, i = 1, \ldots, N \).

In order to derive non-linear decision functions, the so-called kernel trick is exploited. That is, it is assumed that the training vectors \( x_i \) are non-linearly mapped to an arbitrary-dimensional feature space \( \mathcal{F} \) (usually having the properties of Hilbert spaces [31, 39]) by employing a function \( \phi(\cdot) : x_i \in \mathbb{R}^D \rightarrow \phi(x_i) \in \mathcal{F} \).
In \( F \), dot products between training vectors are defined by a kernel function \( \kappa(\cdot, \cdot) \) and are stored in the so-called kernel matrix \( K \in \mathbb{R}^{N \times N} \). Thus, (3) can be given in the form:

\[
max_{\alpha} \ 1^T \alpha - \frac{1}{2} (\alpha \circ y)^T K (\alpha \circ y),
\]

where \( y \in \mathbb{R}^N \) is a vector containing the binary labels \( y_i, \ i = 1, \ldots, N \) and \( \circ \) denotes the Hadamard (element-wise) product operator.

2.2. Multi-class SVM classifier

An extension of the binary SVM classifier in multi-class classification problems that solves a combined optimization problem has been proposed in [47, 48]. The optimization problem solved in order to define \( K \) hyperplanes described by the vectors \( w_k, \ k = 1, \ldots, K \) has the following form:

\[
\min_{w_k, b_k} \sum_{k=1}^{K} \frac{1}{2} w_k^T w_k + c \sum_{i=1}^{N} \sum_{k \neq l} \xi_i^k\]

subject to the constraints:

\[
w_i^T x_i + b_i \geq w_k^T x_i + b_k + 2 - \xi_i^k, \quad \xi_i^k \geq 0, \quad i = 1, \ldots, N, \ k \neq l. \quad (6)
\]

By formulating the dual to (5) optimization problem, the following equivalent problem is obtained:

\[
max_{\alpha_k} \sum_{k=1}^{K} \sum_{i=1}^{N} \sum_{j=1}^{N} \left( \alpha_i^k \alpha_j^l - \frac{1}{2} \alpha_i^k \alpha_j^k - \frac{1}{2} \alpha_i \alpha_j \right) x_i^T x_j + 2 \sum_{k=1}^{K} \sum_{i=1}^{N} \alpha_i^k, \quad (7)
\]
subject to the constraints:

$$\sum_{i=1}^{N} \alpha_{i}^{k} = \sum_{i=1}^{N} c_{i}^{k} \alpha_{i}, \quad k = 1, \ldots, K \quad (8)$$

and

$$0 \leq \alpha_{i}^{k} \leq c, \quad \alpha_{i}^{l} = 0, \quad i = 1, \ldots, N, \quad k \neq l. \quad (9)$$

In the above, $\alpha_{i}^{k}, \ i = 1, \ldots, N, \ k = 1, \ldots, K$ are the Lagrange multipliers and $c_{i}^{k}, \alpha_{i}$ are variables defined as:

$$\alpha_{i} = \sum_{k=1}^{K} \alpha_{i}^{k}, \quad c_{i}^{k} = 1, \ if \ l_{i} = k \ and \ c_{i}^{k} = 0, \ if \ l_{i} \neq k. \quad (10)$$

For the derivation of the solution (7), the reader can refer to the details in [47, 48] and [43]. An important property of (7) is that it is a quadratic function in terms of $\alpha$ and, thus, its global optimal solution can be obtained. In addition, the kernel trick can also be exploited in order to define non-linear decision functions for multi-class classification. It has also been shown in [47, 48] that the binary SVM classifier described in Section 2.1 is a special case of the multi-class SVM classifier for $K = 2$.

2.3. Graph Embedding

The Graph Embedding framework [49] assumes that the training data $x_{i}, \ i = 1, \ldots, N$ are employed in order to form the vertex set of an undirected weighted graph $G = \{X, V\}$, where $X = [x_{1}, \ldots, x_{N}]$ and $V \in \mathbb{R}^{N \times N}$ is a similarity matrix whose elements denote the relationships between the graph vertices $x_{i}$.
Furthermore, a penalty graph $G^p = \{X, V^p\}$ can be defined, whose weight matrix $V^p \in \mathbb{R}^{N \times N}$ penalizes specific relationships between the graph vertices $x_i$.

In the case of linear data projections\(^2\), the data $x_i \in \mathbb{R}^D$ are projected to a low-dimensional feature space $\mathbb{R}^d$, $d < D$, by applying a linear transformation, i.e., $s_i = W^T x_i$. This transformation is obtained by optimizing for:

$$
W^* = \arg\min_{tr(W^T X C X^T W) = c} \sum_{i,j=1}^{N} \|W^T x_i - W^T x_j\|_2^2 V_{ij}
= \arg\min_{tr(W^T X C X^T W) = c} tr(W^T X L X^T W),
$$

(11)

where $tr(\cdot)$ is the trace operator and $L \in \mathbb{R}^{N \times N}$ is the so-called graph Laplacian matrix defined as $L = D - V$, where $D$ is the diagonal degree matrix having elements $D_{ii} = \sum_{j=1}^{N} V_{ij}$. $C \in \mathbb{R}^{N \times N}$ is the graph Laplacian matrix of $G^p$, that is $C = L^p = D^p - V^p$. In the case where no penalty criteria are taken into account, $C$ can be set equal to a constraint matrix, e.g. a diagonal matrix for scale normalization, that is used in order to avoid trivial solutions.

The solution of (11) is obtained by solving the generalized eigenvalue decomposition problem $S_i v = \lambda S_p v$, where $S_i = XLX^T$ is a matrix expressing the data relationships that are subject to minimization and $S_p = XCX^T$ is a matrix expressing the data relationships that are subject to maximization. That is, the columns of the transformation matrix $W$ are formed by the eigenvectors of the matrix $S = S_p^{-1} S_i$ corresponding to the $d$ minimal eigenvalues $\lambda_i$. In

\(^2\)Non-linear data projections can also be obtained by exploiting the kernel trick and the Representer Theorem [39, 49].
the case where the matrix $S_p$ is singular, a regularized version is exploited, i.e. $\tilde{S}_p = S_p + rI$, and eigenanalysis is performed to the matrix $S = \tilde{S}_p^{-1}S_i$. $r$ is a parameter that is used in order to exploit the dominant diagonal property of non-singular matrices \(^3\).

From the above, it can be seen that the matrix $S = \tilde{S}_p^{-1}S_i$ can be employed in order to describe both intrinsic and penalty relationships between the training data.

### 2.4. SVM classifiers exploiting intrinsic graphs

In order to exploit intrinsic graph structures in binary SVM-based classification, the following optimization problem was proposed in [1]:

$$
\min_{w,b} \frac{1}{2} w^T \tilde{S}_i w + c \sum_{i=1}^{N} \xi_i,
$$

subject to the constraints:

$$
y_i(w^T x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, N,
$$

where $\tilde{S}_i \in \mathbb{R}^{D \times D}$ is a matrix describing the properties of the training data that are subject to minimization and is defined as $\tilde{S}_i = I + \lambda XLX^T$. $L$ is the Laplacian matrix of the intrinsic graph defined under the Graph Embedding framework [49], as previously described, and $\lambda > 0$ is a regularization parameter. It should be noted here that the optimization problem in (12) is a generalization of the opti-\(^3\)A value of $r = 0.01$ or $r = 0.001$ is usually used.
mization problems proposed in [42, 50], where only the within-class scatter of the training data is considered, i.e.:

$$S_w = \sum_{k=1}^{K} \sum_{i, l_i = k} (x_i - m_k)(x_i - m_k)^T = XL_wX^T,$$  \hspace{1cm} (14)$$

where $L_w = I - \sum_{k=1}^{K} \frac{1}{N_k} (e_k)(e_k)^T$, $m_k \in \mathbb{R}^D$ is the mean vector of class $k$ formed by $N_k$ samples and $e_k \in \mathbb{R}^N$ is a vector with elements equal to $e^k_i = 1$ if $l_i = k$ and $e^k_i = 0$ if $l_i \neq k$.

For multi-class classification, the following optimization problem was proposed in [21, 22]:

$$\min_{w_k, b_k} \sum_{k=1}^{K} \frac{1}{2} w_k^T S_w w_k + c \sum_{i=1}^{N} \sum_{k \neq l_i} \xi^k_i$$

subject to the constraints:

$$w_k^T x_i + b_k \geq w_{l_i}^T x_i + b_{l_i} + 2 - \xi^k_i, \quad \xi^k_i \geq 0, \quad i = 1, \ldots, N, \quad k \neq l_i.$$  \hspace{1cm} (16)$$

For the extension of the multi-class optimization problem (15) to non-linear decision functions, two-step processes were proposed. Specifically, in [21], the training data $x_i$ are projected to $\tilde{x}_i$ by using $\tilde{x}_i = S^{-\frac{1}{2}}_w x_i$ and the nonlinear version of the standard multi-class SVM (5) is subsequently solved by using $\tilde{x}_i$. In [22], the training data $x_i$ are non-linearly mapped to the feature space determined by applying kernel PCA, and subsequently the linear optimization problem (15) is
solved in that space.

3. Proposed Method

In this Section, we propose a new optimization problem for SVM-based multi-class classification that is able to exploit training data relationships described in both intrinsic and penalty graphs. Specifically, we propose the following optimization problem optimizing (a regularized version of) the discrimination criterion used in Graph Embedding subject to the SVM separability constraints for multi-class classification:

$$\min_{w_k, b_k} \sum_{k=1}^{K} \frac{1}{2} w_k^T w_k + c \sum_{i=1}^{N} \sum_{k \neq l_i} \xi_{i}^{k} + \sum_{k=1}^{K} \frac{\lambda}{2} w_k^T S w_k$$  \hspace{1cm} (17)

subject to the constraints:

$$w_{i}^T x_i + b_i \geq w_{k}^T x_i + b_k + 2 - \xi_{i}^{k}, \quad \xi_{i}^{k} \geq 0, \quad i = 1, \ldots, N, \quad k \neq l_i.$$  \hspace{1cm} (18)

In (17), $S$ is a matrix expressing a combination of intrinsic and penalty training data relationships, as described in Section 2.3, i.e. $S = \tilde{S}_{p}^{-1} S_{i}$, and $\lambda \geq 0$ is a parameter denoting the (relative) importance of the two regularization terms. In the following we provide the solutions of the proposed optimization problem in both the linear and non-linear case, starting from the linear one.
3.1. Linear Case

The equivalent dual optimization problem to (17) subject to the constraints in (18), is the following:

\[ D = \frac{1}{2} \sum_{k=1}^{K} \mathbf{w}_k^T (\mathbf{I} + \lambda \mathbf{S}) \mathbf{w}_k + c \sum_{i=1}^{N} \sum_{k \neq l_i} \xi_i^k - \sum_{k=1}^{K} \sum_{i=1}^{N} \beta_i^k \xi_i^k \]

\[ - \sum_{k=1}^{K} \sum_{i=1}^{N} \alpha_i^k [(\mathbf{w}_{l_i} - \mathbf{w}_k)^T \mathbf{x}_i + b_{l_i} - b_k - 2 + \xi_i^k] \]

(19)

with the constraints:

\[ \alpha_i^k \geq 0, \quad \beta_i^k \geq 0, \quad \xi_i^k \geq 0, \quad i = 1, \ldots, N, \quad k \neq l_i. \]  

(20)

By determining the saddle points of \( D \) with respect to \( \mathbf{w}_k, b_k \) and \( \xi_i^k \), we obtain:

\[ \nabla D|_{\mathbf{w}_k} = 0 \Rightarrow \mathbf{w}_k = (\mathbf{I} + \lambda \mathbf{S})^{-1} \sum_{i=1}^{N} (\alpha_i^k \mathbf{c}_i^k - \alpha_i^k) \mathbf{x}_i, \]  

(21)

\[ \nabla D|_{b_k} = 0 \Rightarrow \sum_{i=1}^{N} \alpha_i^k - \sum_{i=1}^{N} \alpha_i^k \mathbf{c}_i^k = 0, \]  

(22)

\[ \nabla D|_{\mathbf{x}_i} = 0 \Rightarrow c = \alpha_i^k + \beta_i^k, \]  

(23)

with the constraints \( 0 \leq \alpha_i^k \leq c \). Substituting (21), (22) and (23) in (19), we obtain:

\[ D = \sum_{k=1}^{K} \sum_{i=1}^{N} \sum_{j=1}^{N} q_{ij}^k \mathbf{x}_i^T (\mathbf{I} + \lambda \mathbf{S})^{-1} \mathbf{x}_j + 2 \sum_{k=1}^{K} \sum_{i=1}^{N} \alpha_i^k, \]  

(24)
where \( q_{ij}^k = \alpha_i^k \alpha_j^l - \frac{1}{2} \alpha_i^k \alpha_j^l - \frac{1}{2} \alpha_i \alpha_j c_{ij} \), with the constraints:

\[
\sum_{i=1}^{N} \alpha_i^k = \sum_{i=1}^{N} c_i^k \alpha_i, \quad k = 1, \ldots, K
\]  

(25)

and

\[
0 \leq \alpha_i^k \leq c, \quad \alpha_i^l = 0, \quad i = 1, \ldots, N, \quad k \neq l_i,
\]  

(26)

which is a quadratic optimization problem in terms of \( \alpha \).

By comparing (24) and (7), we can observe that the two optimization problems are similar. In order to exploit optimized multi-class SVM implementations solving the problem in (7), e.g. [24], we proceed as follows. Let us define the eigenvalue decomposition \((I + \lambda S) = U \Lambda U^T\), where \( \Lambda \in \mathbb{R}^{D \times D} \) is a diagonal matrix containing the eigenvalues \( \lambda_i, i = 1, \ldots, D \) in a descending order and \( U \in \mathbb{R}^{D \times D} \) is an orthogonal matrix containing the eigenvectors corresponding to the eigenvalues \( \lambda_i \) as columns. Then, we can define a transformation matrix \( P \) satisfying \((I + \lambda S)^{-1} = U \Lambda^{-\frac{1}{2}} \Lambda^{-\frac{1}{2}} U^T = P^T P\). After the calculation of the matrix \( P = \Lambda^{-\frac{1}{2}} U^T \), the application of the optimization problem (7) on the data \( \tilde{x}_i = Px_i, \quad i = 1, \ldots, N \) is equivalent to the application of the optimization problem (24) on the original data \( x_i, \quad i = 1, \ldots, N \).

\( ^4 \)Note that since \( U \) is orthonormal, the matrix \( \tilde{P} = U \Lambda^{-\frac{1}{2}} U^T \) can also be used in the place of \( P \).
3.2. Non-linear Case

In order to obtain non-linear decision functions, we follow the standard kernel approach. That is, we assume that the training vectors $x_i$ are mapped to an arbitrary-dimensional feature space $\mathcal{F}$ (having the properties of Hilbert spaces [31, 39]) by employing a non-linear function $\phi(\cdot) : x_i \in \mathbb{R}^D \rightarrow \phi(x_i) \in \mathcal{F}$. The application of linear classification using (17) in $\mathcal{F}$, corresponds to non-linear classification in the input space $\mathbb{R}^D$.

Let us denote by $\Phi \in \mathbb{R}^{|\mathcal{F}| \times N}$ a matrix (of arbitrary dimensions) containing the training data representations in $\mathcal{F}$. The kernel matrix can be defined as $K = \Phi^T \Phi$.

Based on the Representer Theorem [31, 39], we can also define:

$$ w_k = \sum_{i=1}^{N} \gamma_i^k \phi(x_i) = \Phi \gamma_k, \quad (27) $$

where $\gamma_k \in \mathbb{R}^N$ is a vector containing the reconstruction weights of $w_k$ with respect to $\phi(x_i), i = 1, \ldots, N$. Finally, the matrix (of arbitrary dimensions) $S \in \mathbb{R}^{|\mathcal{F}| \times |\mathcal{F}|}$ can be expressed as follows:

$$ S = (\Phi L_p \Phi^T + rI)^{-1}(\Phi L_i \Phi^T) = \left( \frac{1}{r}I - \frac{1}{r^2} \Phi (L_p^{-1} + \frac{1}{r}K)^{-1} \Phi^T \right) (\Phi L_i \Phi^T) $$

$$ = \frac{1}{r} \Phi L_i \Phi^T - \frac{1}{r^2} \Phi (L_p^{-1} + \frac{1}{r}K)^{-1} KL_i \Phi^T \quad (28) $$

Using (27) and (28), the equivalent to (17) subject to the constraints in (18),
dual optimization problem is the following:

\[
D = \frac{1}{2} \sum_{k=1}^{K} \gamma_k^T \Theta \gamma_k + c \sum_{i=1}^{N} \sum_{k \neq l_i} \xi_k^i - \sum_{k=1}^{K} \sum_{i=1}^{N} \beta_k^i \xi_k^i \\
- \sum_{k=1}^{K} \sum_{i=1}^{N} \alpha_i^k \left[ (\gamma_i - \gamma_k)^T k_i + b_i - b_k - 2 + \xi_k^i \right],
\]

(29)

with the constraints:

\[
\alpha_i^k \geq 0, \quad \beta_i^k \geq 0, \quad \xi_i^k \geq 0, \quad i = 1, \ldots, N, \quad k \neq l_i.
\]

(30)

In (29), \( \Theta = K + \frac{\lambda}{r} KL_i K - \frac{\lambda}{r} K (L_p^{-1} + \frac{1}{r} K)^{-1} KL_i K \) and \( k_i \in \mathbb{R}^N \) is a vector containing the values \( k_{i,j} = \phi(x_j)^T \phi(x_i) \).

By determining the saddle points of \( D \) with respect to \( \gamma_k, b_k \) and \( \xi_i^k \), we obtain:

\[
\nabla D|_{\gamma_k} = 0 \Rightarrow \gamma_k = \Theta^{-1} \sum_{i=1}^{N} (\alpha_i c_i^k - \alpha_i^k) k_i, \quad (31)
\]

\[
\nabla D|_{b_k} = 0 \Rightarrow \sum_{i=1}^{N} \alpha_i^k - \sum_{i=1}^{N} \alpha_i c_i^k = 0, \quad (32)
\]

\[
\nabla D|_{x_i^k} = 0 \Rightarrow c = \alpha_i^k + \beta_i^k, \quad (33)
\]

with the constraints \( 0 \leq \alpha_i^k \leq c \).

Substituting (31), (32) and (33) in (29), we obtain:

\[
D = \sum_{k=1}^{K} \sum_{i=1}^{N} \sum_{j=1}^{N} q_{ij}^k k_i^T (\Theta^{-1})^T k_j + 2 \sum_{k=1}^{K} \sum_{i=1}^{N} \alpha_i^k,
\]

(34)
where \( q_{ij}^k = \alpha_i^k \alpha_j^k - \frac{1}{2} \alpha_i^k \alpha_j^k - \frac{1}{2} \alpha_i^k \alpha_j^k \), with the constraints:

\[
\sum_{i=1}^{N} \alpha_i^k = \sum_{i=1}^{N} c_i^k \alpha_i, \quad k = 1, \ldots, K
\]  

(35)

and

\[
0 \leq \alpha_i^k \leq c_i, \quad \alpha_i^{l_i} = 0, \quad i = 1, \ldots, N, \quad k \neq l_i
\]  

(36)

which is a quadratic optimization problem in terms of \( \alpha \).

Similar to the linear case described in subsection 3.1, we can exploit optimized multi-class SVM implementations, e.g. [24], by applying the multi-class SVM (7) in a transformed feature space. Specifically, the solution of (34) is equivalent to the solution of (7), by exploiting the modified kernel matrix \( \tilde{K} = K(\Theta^{-1})^T K = (I + \frac{\lambda}{\tau} KL_i - \frac{\lambda}{\tau^2} K(L_p^{-1} + \frac{\lambda}{\tau} K)^{-1} KL_i)^T K \). Here we should note that the calculation of \( \tilde{K} \) requires the inversion of two \( N \times N \) matrices, which is computationally intensive in large-scale classification problems.

3.3. Discussion

Here we discuss the connection of the proposed method with that of [42, 50, 1, 35, 21, 22]. Specifically, we show that all these methods are special cases of the proposed approach. As an additional contribution, we show that the methods in [21, 22] are direct extensions of [42, 50] to multi-class classification. Moreover, we discuss the connection and the differences between the proposed approach with recent works combining feature learning with classification.

First, we derive a binary SVM classifier as a special case of the proposed
method for $K = 2$. In this case, if we set $l_i \in \{0, 1\}$, the constraints in (18) become:

$$w_{l_i}^T x_i + b_{l_i} \geq w_{1-l_i}^T x_i + b_{1-l_i} + 2 - \xi_i^{1-l_i}. \quad (37)$$

The separation of the two classes by the maximum of two linear functions is equivalent to separating them by a single hyperplane with the constraints:

$$(w_{l_i} - w_{1-l_i})^T x_i + (b_{l_i} - b_{1-l_i}) \geq 2 - \xi_i^{1-l_i}. \quad (38)$$

Setting $w = w_{l_i} - w_{1-l_i}$, $b = b_{l_i} - b_{1-l_i}$ and $\xi_i = 1 - \xi_i^{1-l_i}$, the proposed optimization problem is equivalent to the following one:

$$\min_{w, b} \frac{1}{2} w^T w + c \sum_{i=1}^N \xi_i + \frac{\lambda}{2} w^T S w \quad (39)$$

subject to the constraints:

$$w^T x_i + b \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, N, \quad k \neq l_i. \quad (40)$$

Thus, the binary SVM (1) is a special case of the proposed method (39), for $\lambda = 0$. In order to derive the binary SVM formulation exploiting only intrinsic graph structures (12), we follow the standard Graph Embedding approach and let the matrix $S_p$ be equal to a constraint matrix, i.e. in the simplest case $S_p = I$. In a similar way, the multi-class SVM problem in (5) is a special case of the proposed approach for $\lambda = 0$. The multi-class formulations exploiting the within-
class variance of the training data (15) correspond to the case where \( S_p = I \) and \( S_i = S_w \) (14). Finally, (39) is a regularized extension of the methods in [42, 50] for \( S_p = I \) and \( S_i = S_w \) (14). Thus, [42, 50] are special cases of (15) for \( K = 2 \).

The major advantage of the proposed approach is that it is able to exploit data relationships described in both intrinsic and penalty graphs. In this paper, we evaluate the performance achieved by exploiting the graph structures used in LDA, MDA and LFDA. However, it should be noted that any graph structure defined within the Graph Embedding framework could be exploited. This fact gives us the opportunity to exploit information related to specific problems and define and exploit new graph structures as well.

LDA uses the following graph weights:

\[ V_{ij} = \begin{cases} \frac{1}{N_i}, & l_j = l_i, \\ 0, & \text{otherwise}, \end{cases} \quad V_{ij}^p = \begin{cases} \frac{1}{N} - \frac{1}{N_i}, & l_j = l_i, \\ \frac{1}{N}, & \text{otherwise}, \end{cases} \] (41)

for the intrinsic and penalty graphs, respectively. MDA uses graph weights defined by:

\[ V_{ij} = \begin{cases} 1, & l_i = l_j \text{ and } x_j \in N_i, \\ 1, & l_i = l_j \text{ and } x_i \in N_j, \\ 0, & \text{otherwise}, \end{cases} \quad V_{ij}^p = \begin{cases} 1, & l_i \neq l_j \text{ and } x_j \in N_i, \\ 1, & l_i \neq l_j \text{ and } x_i \in N_j, \\ 0, & \text{otherwise}. \end{cases} \] (42)

\( N_i \) denotes the neighborhood of sample \( x_i \). In all our experiments we have used 5-NN graphs. While larger neighborhoods can also be exploited, we have ex-
experimentally found that by using small neighborhoods, local relationships can be better exploited. Finally, LFDA uses graph weights defined by:

\[
V_{ij} = \begin{cases} 
\frac{v_{ij}}{N_i}, & l_j = l_i, \\
0, & \text{otherwise.}
\end{cases}
\]

\[
V_{ij}^p = \begin{cases} 
\frac{v_{ij}}{N_i} \left(1 - \frac{1}{N_i}\right), & l_j = l_i, \\
\frac{1}{N}, & \text{otherwise.}
\end{cases}
\]

(43)

where \(v_{ij}\) is a measure of the similarity between \(x_i\) and \(x_j\). In our experiments, we used the heat kernel function (also known as RBF kernel). It should be noted here that, for non-linear classification, the similarities \(v_{ij}\) and the data neighborhoods \(N_i\) should be calculated in \(F\), i.e. by using \(\phi(x_i), i = 1, \ldots, N\).

Recently, feature learning has been combined with ensemble-based and multi-class classification [53, 54, 33]. The idea in these methods is to jointly learn a data mapping from the input space to an optimal (usually low-dimensional) feature space, where classification is enhanced. In [53] this idea has been combined with multiple binary regression models, which are subsequently combined following the Error-Correcting Output Codes (ECOC) framework in order to form a multi-class model. Iterative (gradient-based) optimization is employed to this end, since the method exploits a non-convex optimization problem. The standard SVM formulation was combined with ECOC in [54], where SVM classifier is treated as a black box binary classifier the outputs of which are transformed to class probability values. An iterative learning process is subsequently applied in order to determine the optimal ECOC parameters. Finally, [33] followed a multi-class SVM formulation, that iteratively optimizes data representation and SVM parameters. While these methods are able to jointly learn an optimal data representation and
the parameters of a multi-class classifier scheme, they cannot incorporate a priori knowledge described in the form of graphs (expressing pairwise relationships between the data).

4. Experiments

In this Section, we describe experiments conducted in order to evaluate the performance of the proposed approach. We compare the performance of the proposed classifiers with that of the standard SVM classification and the SVM classifiers exploiting intrinsic graphs in human action recognition. Subsequently, we apply the proposed classifiers in facial image and standard classification problems. In human action recognition, we employ three active benchmark datasets, i.e. the Hollywood2, the Olympic sports and the recently introduced Hollywood 3D datasets. In facial image classification, we employ two facial image datasets, i.e. the JAFFE [28] and ORL [37] datasets. Finally, we applied the proposed SVM classifier on four standard classification problems coming from the machine learning repository of the University of California Irvine (UCI) [2]. In all the experiments we compare the performance of the proposed SVM classifier using the graph structures described in Subsection 3.3, with the standard SVM classifier, the Simple SVM (SimSVM) classifier of [10]\(^5\), the Minimum Class Variance SVM (MCVSVM) classifier of [22]\(^6\) and the Graph Embedded SVM (GESVM)

---

\(^5\)For the implementation of SimSVM, we used the multiplicative update rules of [40].

\(^6\)Note that as has been previously described the classifier of [22] is an extension of the MCVSVM classifiers of [42, 50] for multi-class classification.
classifier of [1] with the same graph structures used for the proposed classifiers\textsuperscript{7}.

4.1. Experiments in Human Action Recognition

The Hollywood\textsuperscript{2} dataset \cite{29} consists of 1707 videos depicting 12 actions. The videos have been collected from 69 different Hollywood movies. We used the standard training-test split provided by the database (823 videos are used for training and performance is measured in the remaining 884 videos). Training and test videos come from different movies. The performance is evaluated by computing the average precision (AP) for each action class and reporting the mean AP over all classes (mAP). This is due to the fact that a video may depict more than one action. Video frames from the Hollywood\textsuperscript{2} dataset are illustrated in Figure 1.

The Olympic sports dataset \cite{32} consists of 783 videos depicting athletes practicing 16 sports. We used the standard training-test split provided by the database (649 videos are used for training and performance is measured in the remaining

\textsuperscript{7}Note that the GESVM classifier of [1] includes as special cases the classifiers of [35, 21] when using different intrinsic graph structures.
134 videos). The performance is evaluated by computing the mean Average Precision (mAP) over all classes. In addition, since each video depicts only one action, performance can be measured by computing the classification rate (CR) of a given method. Video frames from the Olympic sports dataset are illustrated in Figure 2.

The Hollywood 3D dataset [8] consists of 951 video pairs (left and right channel) depicting 13 actions and a ‘no action’ class collected from Hollywood movies. We used the left channel for each video and the standard (balanced) training-test split provided by the database (643 videos are used for training and performance is measured in the remaining 308 videos). Training and test videos come from different movies. The performance is evaluated by computing both the mean AP over all classes (mAP) and the classification rate (CR) metrics. Video frames from the Hollywood 3D dataset are illustrated in Figure 3.

In our first set of experiments, we applied the competing methods by fol-
Following the experimental protocols provided by the databases. That is, for each dataset, we formulate $K$ One-Versus-Rest classification problems, where $K = 12$, $K = 16$ and $K = 13$ for the Hollywood2, the Olympic sports and Hollywood 3D databases, respectively. After training the corresponding binary classifiers by using the training videos, we introduce the test videos and measure the performance of each classifier by calculating the mAP metric. We use the state-of-the-art video representation proposed in [46] that describes a video by using HOG, HOF, MBHx, MBHy and (normalized) Trajectory descriptors evaluated on the trajectories of densely sampled interest points. After descriptor calculation, the video is represented by $V = 5$ Bag-of-Words (BoW)-based representations, i.e. one BoW-based representation per descriptor type. We follow [46] and use 4000 codewords for each BoW representation. Classification is performed by employing the RBF-$\chi^2$ kernel which has been found to outperform other choices for BoW-based representations [23]. Different descriptor types are combined by following a multi-channel approach [52] $[K]_{ij} = exp \left( -\sum_{v=1}^{V} \sum_{d=1}^{D} \frac{1}{2 \Delta v} \frac{(x_{vd}^v - x_{jd}^v)^2}{x_{vd}^v + x_{jd}^v} \right)$.
where $A^v$ is set to the mean value of the $\chi^2$ distances between the training vectors corresponding to descriptor $v$. The values of the parameters $c$ and $\lambda$ have been chosen by applying grid search using the values $c = 10^{-3,\ldots,3}$ and $\lambda = 10^{-3,\ldots,3}$ (in log scale). In all our experiments we have used the optimized SVM implementations in [4, 24].

Experimental results are illustrated in Figure 4 and in Table 1. The exploitation of data relationships described in intrinsic graph structures clearly enhances the performance of SVM classifier. Exploitation of the information described in both intrinsic and penalty graphs further enhances performance, since the proposed methods achieve the best performance in all three databases.

In Table 2, we also compare the performance obtained by applying the proposed classifiers on the BoW-based video representation exploiting the Improved Dense Trajectory-based video description with that of some recently proposed action recognition methods. It should be noted here that most of them are not directly comparable to our results due to the use of different features and representations. The most comparable ones are the methods in [46, 14, 15, 16] (starred

### Table 1: Performance (mAP) in human action recognition.

<table>
<thead>
<tr>
<th></th>
<th>Hollywood2</th>
<th>Olympic sports</th>
<th>Hollywood 3D</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>61.41%</td>
<td>82.77%</td>
<td>29.45%</td>
</tr>
<tr>
<td>MCVSVM [22]</td>
<td>65.98%</td>
<td>84.94%</td>
<td>30.31%</td>
</tr>
<tr>
<td>GESVM [1] with (41)</td>
<td>65.74%</td>
<td>84.86%</td>
<td>30.29%</td>
</tr>
<tr>
<td>GESVM [1] with (42)</td>
<td>66.06%</td>
<td>84.96%</td>
<td>31.11%</td>
</tr>
<tr>
<td>GESVM [1] with (43)</td>
<td>66.03%</td>
<td>84.99%</td>
<td>30.29%</td>
</tr>
<tr>
<td>Proposed with (41)</td>
<td>67.51%</td>
<td>87.83%</td>
<td>32.86%</td>
</tr>
<tr>
<td>Proposed with (42)</td>
<td>67.48%</td>
<td>87.82%</td>
<td>33.13%</td>
</tr>
<tr>
<td>Proposed with (43)</td>
<td>67.5%</td>
<td>88.12%</td>
<td>33.23%</td>
</tr>
</tbody>
</table>
Figure 4: Performance on action recognition datasets: (top) Hollywood2, (middle) Olympic Sports and (bottom) Hollywood 3D.
Table 2: Comparison of our results with some state-of-the-art methods on the Hollywood2, Olympic Sports and Hollywood 3D datasets.

<table>
<thead>
<tr>
<th>Method</th>
<th>Hollywood2</th>
<th>Olympic sports</th>
<th>Hollywood 3D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method [14]$^*$</td>
<td>-</td>
<td>-</td>
<td>29.28%</td>
</tr>
<tr>
<td>Method [13]</td>
<td>-</td>
<td>-</td>
<td>30.52%</td>
</tr>
<tr>
<td>Method [13]</td>
<td>-</td>
<td>-</td>
<td>30.52%</td>
</tr>
<tr>
<td>Method [9]</td>
<td>-</td>
<td>-</td>
<td>36.9%</td>
</tr>
<tr>
<td>Method [41]</td>
<td>48.1%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Method [38]</td>
<td>59.6%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Method [44]</td>
<td>61.9%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Method [6]</td>
<td>-</td>
<td>82.7%</td>
<td>-</td>
</tr>
<tr>
<td>Method [18]</td>
<td>59.5%</td>
<td>80.6%</td>
<td>-</td>
</tr>
<tr>
<td>Method [17]</td>
<td>62.3%</td>
<td>83.2%</td>
<td>-</td>
</tr>
<tr>
<td>Method [7]</td>
<td>-</td>
<td>85.5%</td>
<td>-</td>
</tr>
<tr>
<td>Method [34]</td>
<td>63.3%</td>
<td>89%</td>
<td>-</td>
</tr>
<tr>
<td>Method [15]$^*$</td>
<td>61.69%</td>
<td>88.89%</td>
<td>-</td>
</tr>
<tr>
<td>Method [16]$^*$</td>
<td>62.3%</td>
<td>89.74%</td>
<td>31.79%</td>
</tr>
<tr>
<td>Method [46] - BoWs$^*$</td>
<td>62.2%</td>
<td>83.3%</td>
<td>-</td>
</tr>
<tr>
<td>Method [46] - FVs</td>
<td>64.3%</td>
<td>91.1%</td>
<td>-</td>
</tr>
<tr>
<td><strong>Proposed method</strong></td>
<td><strong>67.51%</strong></td>
<td><strong>88.12%</strong></td>
<td><strong>33.23%</strong></td>
</tr>
</tbody>
</table>

In Table 2) exploiting the same video representation. Nevertheless, it can be seen that the proposed approach can achieve state-of-the-art performance. It is worth noting here that most of the methods listed in Table 2 employ standard (binary) SVM classification. Thus, we expect that the application of the proposed classifiers would enhance their performance.

In our second experiment we apply multi-class classification on the Olympic sports and Hollywood 3D databases. Experimental results are illustrated in Table 3. The exploitation of data relationships described in intrinsic graph structures clearly enhances the performance of SVM also in multi-class classification. Similarly to the binary classification, the exploitation of the information described in both intrinsic and penalty graphs further enhances performance, since the proposed methods achieve the best performance in both databases. For comparison
reasons, we have also applied the Graph Embedded Kernel Extreme Learning Machine (GEKELM) classifier proposed in [16] using the graphs defined in (41)-(43) and include the results in Table 3. It is worth noting here that, while GEKELM exploits a similar regularization term for Extreme Learning Machine (ELM)-based Single-hidden Layer Feedforward Neural (SLFN) network training, the resulting classifier is in essence an OVR classification scheme. Thus, it is not able to take into account possible correlations between the various two-class classification problems solved in order to achieve multi-class classification.

<table>
<thead>
<tr>
<th></th>
<th>Olympic sports</th>
<th>Hollywood 3D</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>73.88%</td>
<td>27.48%</td>
</tr>
<tr>
<td>SimSVM [10]</td>
<td>70.15%</td>
<td>26.16%</td>
</tr>
<tr>
<td>MCVSVM [22]</td>
<td>75.37%</td>
<td>28.14%</td>
</tr>
<tr>
<td>GESVM [1] with (41)</td>
<td>75.37%</td>
<td>28.14%</td>
</tr>
<tr>
<td>GESVM [1] with (42)</td>
<td>75.37%</td>
<td>28.14%</td>
</tr>
<tr>
<td>GESVM [1] with (43)</td>
<td>74.63%</td>
<td>28.14%</td>
</tr>
<tr>
<td>GEKELM [16] with (41)</td>
<td>75.29%</td>
<td>28.3%</td>
</tr>
<tr>
<td>GEKELM [16] with (42)</td>
<td>73.89%</td>
<td>28.3%</td>
</tr>
<tr>
<td>GEKELM [16] with (43)</td>
<td>75.29%</td>
<td>28.3%</td>
</tr>
<tr>
<td>Proposed with (41)</td>
<td>76.12%</td>
<td>31.79%</td>
</tr>
<tr>
<td>Proposed with (42)</td>
<td>76.12%</td>
<td>31.79%</td>
</tr>
<tr>
<td>Proposed with (43)</td>
<td>76.12%</td>
<td>31.79%</td>
</tr>
</tbody>
</table>

4.2. Experiments in Face Recognition

In our third set of experiments we have applied multi-class classification in two facial image datasets, i.e. the JAFFE [28] and ORL [37] datasets. The JAFFE dataset consists of 210 facial images depicting 10 Japanese female persons. Each person is depicted in 3 images for each expression. The ORL dataset consists of 400 facial images depicting 40 persons. The images were captured at different
times and in different conditions, in terms of lighting, facial expressions (smiling/not smiling) and facial details (open/closed eyes, with/without glasses). Facial images were taken in frontal position with a tolerance for face rotation and tilting up to 20 degrees. Facial images from the JAFFE and ORL datasets are illustrated in Figures 5 and 6, respectively.

In order to make maximal use of the available data and produce average classification rate results, we employ a variant of the leave-one-out cross-validation approach. That is, in each experiment we randomly partition the data belonging to each class of the classification problem at hand in five sets. Each of the five sets contains 20% of the class samples. In each cross-validation round, one set per class is used as a test set, while the remaining sets form the training set. Five cross-validation rounds are conducted, one per each test set index. The classification accuracy of each classifier is subsequently measured for that experiment. We perform five experiments for each database and calculate the mean classification rate in order to measure the performance of the various classifiers. We apply non-linear classification using the RBF kernel function, i.e. $K_{ij} = \exp \left( -\frac{\|x_i - x_j\|^2}{2\sigma^2} \right)$. We set the value of the parameter $\sigma$ equal to the mean Euclidean distance between
the training vectors, which is the natural scaling factor for each database.

Experimental results for facial image classification are illustrated in Figure 7 and in Table 4. Similar to the action recognition case, the exploitation of geometric data relationships enhances the performance of the SVM classifier. Compared to the methods exploiting only geometric data relationships described in intrinsic graphs, the proposed approach further enhances performance by exploiting combined information appearing in both intrinsic and penalty graphs.
Table 4: Performance (CR) in facial image classification problems.

<table>
<thead>
<tr>
<th>Method</th>
<th>JAFFE</th>
<th>ORL</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>82.38%</td>
<td>92.25%</td>
</tr>
<tr>
<td>SimSVM [10]</td>
<td>80.95%</td>
<td>94.5%</td>
</tr>
<tr>
<td>MCVSVM [22]</td>
<td>84.29%</td>
<td>94.5%</td>
</tr>
<tr>
<td>GESVM [1] with (41)</td>
<td>82.38%</td>
<td>94.5%</td>
</tr>
<tr>
<td>GESVM [1] with (42)</td>
<td>84.29%</td>
<td>94.25%</td>
</tr>
<tr>
<td>GESVM [1] with (43)</td>
<td>84.76%</td>
<td>94.5%</td>
</tr>
<tr>
<td>Proposed with (41)</td>
<td>88.1%</td>
<td>98.5%</td>
</tr>
<tr>
<td>Proposed with (42)</td>
<td>87.62%</td>
<td>98.25%</td>
</tr>
<tr>
<td>Proposed with (43)</td>
<td>88.1%</td>
<td>98.5%</td>
</tr>
</tbody>
</table>

4.3. Experiments in Standard Classification Problems

Finally, we have applied multi-class classification in standard classification problems coming from the machine learning repository of the University of California Irvine (UCI) [2]. More precisely, we have used the Iris, Glass, Parkinson and Seeds databases. In order to make maximal use of the available data and produce average classification rate results, we apply the same leave-one-out cross-validation procedure as described above for the facial image classification experiments. Both linear and non-linear cases were tested. For non-linear classification, we used the RBF kernel, where we set the value of the parameter $\sigma$ equal to the mean Euclidean distance between the training vectors, which is the natural scaling factor for each database. 2-dimensional data obtained by applying PCA on the UCI datasets used in our experiments are illustrated in Figure 8.

Experimental results for linear and non-linear classification are illustrated in Figure 9 and in Tables 5 and 6, respectively. Similar to the action recognition and facial image classification cases, the exploitation of geometric data relation-
Figure 8: UCI datasets: a) Glass, b) Heart, c) Iris, d) Parkinson, e) Relax, f) Seeds, g) Spectf and h) Tae).
ships enhances the performance of the SVM classifier. Compared to the methods exploiting only geometric data relationships described in intrinsic graphs, once again, the proposed approach further enhances performance by exploiting the combined information appearing in both intrinsic and penalty graphs. In Tables 7 and 8 we also provide the average number of support vectors used by all the methods for the linear and non-linear cases, respectively. As can be seen, the average number of support vectors for the proposed methods is similar to that of the standard SVM and the SVM solutions exploiting intrinsic graphs.

Table 5: Performance (CR) in standard classification problems for linear classification.

<table>
<thead>
<tr>
<th></th>
<th>Glass</th>
<th>Heart</th>
<th>Iris</th>
<th>Parkinson</th>
<th>Relax</th>
<th>Seeds</th>
<th>Spectf</th>
<th>Tae</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>60.75%</td>
<td>84.44%</td>
<td>93.2%</td>
<td>88.21%</td>
<td>71.43%</td>
<td>96.19%</td>
<td>78.28%</td>
<td>50.99%</td>
</tr>
<tr>
<td>SimSVM [10]</td>
<td>61.21%</td>
<td>83.95%</td>
<td>92.67%</td>
<td>87.18%</td>
<td>72.53%</td>
<td>95.24%</td>
<td>78.28%</td>
<td>49.52%</td>
</tr>
<tr>
<td>MCVSVM [22]</td>
<td>62.62%</td>
<td>84.57%</td>
<td>94.8%</td>
<td>89.74%</td>
<td>71.43%</td>
<td>97.14%</td>
<td>79.4%</td>
<td>51.43%</td>
</tr>
<tr>
<td>GESVM [1] with (41)</td>
<td>62.62%</td>
<td>84.57%</td>
<td>94.93%</td>
<td>89.74%</td>
<td>71.43%</td>
<td>97.14%</td>
<td>79.4%</td>
<td>51.43%</td>
</tr>
<tr>
<td>GESVM [1] with (42)</td>
<td>61.88%</td>
<td>83.83%</td>
<td>94%</td>
<td>88.21%</td>
<td>71.43%</td>
<td>96.67%</td>
<td>79.4%</td>
<td>51.21%</td>
</tr>
<tr>
<td>GESVM [1] with (43)</td>
<td>62.15%</td>
<td>84.2%</td>
<td>94.93%</td>
<td>89.74%</td>
<td>71.43%</td>
<td>97.14%</td>
<td>79.4%</td>
<td>52.32%</td>
</tr>
<tr>
<td>Proposed with (41)</td>
<td>63.08%</td>
<td>84.69%</td>
<td>95.47%</td>
<td>90.26%</td>
<td>71.43%</td>
<td>97.62%</td>
<td>79.4%</td>
<td>52.98%</td>
</tr>
<tr>
<td>Proposed with (42)</td>
<td>64.95%</td>
<td>84.81%</td>
<td>95.47%</td>
<td>90.26%</td>
<td>72.53%</td>
<td>97.62%</td>
<td>79.4%</td>
<td>52.98%</td>
</tr>
<tr>
<td>Proposed with (43)</td>
<td>63.08%</td>
<td>84.81%</td>
<td>95.47%</td>
<td>90.26%</td>
<td>72.53%</td>
<td>97.62%</td>
<td>79.4%</td>
<td>52.98%</td>
</tr>
</tbody>
</table>

Table 6: Performance (CR) in standard classification problems for non-linear classification.

<table>
<thead>
<tr>
<th></th>
<th>Glass</th>
<th>Heart</th>
<th>Iris</th>
<th>Parkinson</th>
<th>Relax</th>
<th>Seeds</th>
<th>Spectf</th>
<th>Tae</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>54.21%</td>
<td>84.07%</td>
<td>93.33%</td>
<td>82.05%</td>
<td>71.43%</td>
<td>92.86%</td>
<td>79.4%</td>
<td>49.89%</td>
</tr>
<tr>
<td>SimSVM [10]</td>
<td>55.51%</td>
<td>84.94%</td>
<td>90%</td>
<td>82.05%</td>
<td>66.67%</td>
<td>93.33%</td>
<td>72.16%</td>
<td>43.71%</td>
</tr>
<tr>
<td>MCVSVM [22]</td>
<td>55.14%</td>
<td>84.32%</td>
<td>94.67%</td>
<td>84.62%</td>
<td>71.79%</td>
<td>93.33%</td>
<td>79.4%</td>
<td>45.7</td>
</tr>
<tr>
<td>GESVM [1] with (41)</td>
<td>55.14%</td>
<td>84.94%</td>
<td>94.67%</td>
<td>84.62%</td>
<td>71.61%</td>
<td>93.33%</td>
<td>80.4%</td>
<td>49.01%</td>
</tr>
<tr>
<td>GESVM [1] with (42)</td>
<td>56.54%</td>
<td>84.94%</td>
<td>94.67%</td>
<td>84.62%</td>
<td>71.61%</td>
<td>93.33%</td>
<td>79.4%</td>
<td>49.01%</td>
</tr>
<tr>
<td>GESVM [1] with (43)</td>
<td>56.54%</td>
<td>84.57%</td>
<td>94%</td>
<td>84.1%</td>
<td>71.79%</td>
<td>93.33%</td>
<td>80.15%</td>
<td>49.67%</td>
</tr>
<tr>
<td>Proposed with (41)</td>
<td>63.08%</td>
<td>84.81%</td>
<td>95.47%</td>
<td>90.26%</td>
<td>72.53%</td>
<td>97.62%</td>
<td>79.4%</td>
<td>52.98%</td>
</tr>
<tr>
<td>Proposed with (42)</td>
<td>65.42%</td>
<td>84.94%</td>
<td>95.33%</td>
<td>86.67%</td>
<td>72.16%</td>
<td>93.81%</td>
<td>80.15%</td>
<td>50.77%</td>
</tr>
<tr>
<td>Proposed with (43)</td>
<td>57.94%</td>
<td>84.94%</td>
<td>94.67%</td>
<td>86.15%</td>
<td>72.16%</td>
<td>93.33%</td>
<td>80.15%</td>
<td>50.55%</td>
</tr>
</tbody>
</table>
Figure 9: UCI datasets: a) performance of linear methods and b) performance of nonlinear methods (from top to the bottom: Glass, Heart, Iris, Parkinson, Relax, Seeds, Spectf and Tae).
Table 7: Number of SVs (average per class) in standard classification problems for linear classification.

<table>
<thead>
<tr>
<th></th>
<th>Glass</th>
<th>Heart</th>
<th>Iris</th>
<th>Parkinson</th>
<th>Relax</th>
<th>Seeds</th>
<th>Spectf</th>
<th>Tae</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>48.5</td>
<td>78.6</td>
<td>26.47</td>
<td>44</td>
<td>86</td>
<td>24.67</td>
<td>93.6</td>
<td>82.07</td>
</tr>
<tr>
<td>SimSVM [10]</td>
<td>28.1</td>
<td>102</td>
<td>25.47</td>
<td>73</td>
<td>72.8</td>
<td>11.8</td>
<td>106.7</td>
<td>40.27</td>
</tr>
<tr>
<td>MCVSVM [22]</td>
<td>50.17</td>
<td>79</td>
<td>26.8</td>
<td>56.8</td>
<td>86.4</td>
<td>62.87</td>
<td>102.6</td>
<td>83.6</td>
</tr>
<tr>
<td>GESVM [1] with (41)</td>
<td>50.17</td>
<td>79</td>
<td>26.8</td>
<td>56.8</td>
<td>86.4</td>
<td>62.87</td>
<td>102.6</td>
<td>83.6</td>
</tr>
<tr>
<td>GESVM [1] with (42)</td>
<td>52.47</td>
<td>84.2</td>
<td>26.67</td>
<td>48.2</td>
<td>90</td>
<td>27.73</td>
<td>102.6</td>
<td>83.8</td>
</tr>
<tr>
<td>GESVM [1] with (43)</td>
<td>51.4</td>
<td>79</td>
<td>26.73</td>
<td>59.8</td>
<td>87.2</td>
<td>63.8</td>
<td>98.2</td>
<td>82.93</td>
</tr>
<tr>
<td>Proposed with (41)</td>
<td>49.37</td>
<td>79.2</td>
<td>27</td>
<td>58.4</td>
<td>86.4</td>
<td>67.07</td>
<td>104</td>
<td>84.4</td>
</tr>
<tr>
<td>Proposed with (42)</td>
<td>52.8</td>
<td>114.8</td>
<td>27</td>
<td>58.2</td>
<td>86.6</td>
<td>28.75</td>
<td>86.6</td>
<td>85.07</td>
</tr>
<tr>
<td>Proposed with (43)</td>
<td>51.23</td>
<td>79.2</td>
<td>27</td>
<td>62</td>
<td>87</td>
<td>72.8</td>
<td>104.2</td>
<td>81.6</td>
</tr>
</tbody>
</table>

Table 8: Number of SVs (average per class) in standard classification problems for non-linear classification.

<table>
<thead>
<tr>
<th></th>
<th>Glass</th>
<th>Heart</th>
<th>Iris</th>
<th>Parkinson</th>
<th>Relax</th>
<th>Seeds</th>
<th>Spectf</th>
<th>Tae</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>57.33</td>
<td>111.8</td>
<td>71.2</td>
<td>72.8</td>
<td>107.4</td>
<td>97.47</td>
<td>88</td>
<td>82.07</td>
</tr>
<tr>
<td>SimSVM [10]</td>
<td>28.53</td>
<td>100.3</td>
<td>40</td>
<td>74.7</td>
<td>72.6</td>
<td>56</td>
<td>106.8</td>
<td>40.27</td>
</tr>
<tr>
<td>MCVSVM [22]</td>
<td>57.37</td>
<td>119.6</td>
<td>80.47</td>
<td>45.8</td>
<td>83.2</td>
<td>112</td>
<td>88</td>
<td>83.6</td>
</tr>
<tr>
<td>GESVM [1] with (41)</td>
<td>57.13</td>
<td>112.2</td>
<td>71.53</td>
<td>72.8</td>
<td>107.4</td>
<td>78.4</td>
<td>104.6</td>
<td>83.6</td>
</tr>
<tr>
<td>GESVM [1] with (42)</td>
<td>50.47</td>
<td>183.4</td>
<td>71.2</td>
<td>60.8</td>
<td>107.8</td>
<td>78.4</td>
<td>88</td>
<td>83.8</td>
</tr>
<tr>
<td>GESVM [1] with (43)</td>
<td>57.13</td>
<td>112.2</td>
<td>71.53</td>
<td>38.8</td>
<td>107.6</td>
<td>78.4</td>
<td>90.2</td>
<td>82.93</td>
</tr>
<tr>
<td>Proposed with (41)</td>
<td>50.2</td>
<td>121.4</td>
<td>80.07</td>
<td>73.2</td>
<td>107</td>
<td>97.47</td>
<td>113.2</td>
<td>84.4</td>
</tr>
<tr>
<td>Proposed with (42)</td>
<td>51.2</td>
<td>112.2</td>
<td>71.53</td>
<td>72.8</td>
<td>107.2</td>
<td>97.87</td>
<td>109.6</td>
<td>85.07</td>
</tr>
<tr>
<td>Proposed with (43)</td>
<td>57.4</td>
<td>117</td>
<td>76.73</td>
<td>76.2</td>
<td>107.4</td>
<td>110.87</td>
<td>83.2</td>
<td>81.5</td>
</tr>
</tbody>
</table>

Overall, the proposed multi-class SVM classifier by exploiting data relationships in a maximum margin formulation is able to exploit geometric information related to the structure of the training data in addition to the positions of the support vectors. As has been shown in subsections 3.1 and 3.2, this is equivalent to the application of maximum margin classification in a transformed feature space, exploiting properties of interest for the data described in the corresponding in-
trinsic and penalty graph structures. By defining appropriate feature spaces, the maximum margin-based classifier is able to achieve a better solution.

4.4. Statistical Significance Analysis of Experimental Results

The Friedman test was used, in order to test the null hypothesis that the competing classifiers perform equally well and the observed differences are merely random [5]. We compare the performance of the four different approaches, i.e. we compare SVM, MCVSVM, GESVM and the proposed approach exploiting both intrinsic and penalty graphs, for nonlinear classification on the 13 employed datasets. For both GESVM and our method we employ the variants exploiting the graph structures (42), which overall provide the best performance when compared to the rest tested graph structures. After ordering the algorithms according to their performance on each data set, the obtained mean ranks are equal to $R_{SVM} = 3.7308$, $R_{MCVSM} = 2.6538$, $R_{GESVM} = 2.5$ and $R_{pGESVM} = 1.154$ for the SVM, MCVSVM, GESVM and the proposed GESVM algorithms, respectively. The overall mean rank is equal to $R_o = 2.5$. The Friedman statistic is equal to $\chi_F^2 = 26.9538$ and $F_F = 26.8506$. With $k = 4$ classifiers and $N = 13$ datasets, $F_F$ is distributed according to an $F$ distribution with $(4 - 1) = 3$ and $(4 - 1) \times (13 - 1) = 36$ degrees of freedom. The critical value of $F(3,36)$ for $\alpha = 0.05$ is 2.86, so we reject the null hypothesis that all the classifiers perform the same.

Following the Nemenyi test for pairwise comparisons [5], we obtain a critical value equal to 2.569 and, thus, the critical difference is equal to $CD =$

37
2.569\sqrt{\frac{k(k+1)}{6N}} = 1.3009. By calculating the differences between the ranks of the three algorithms and the proposed pGESVM, we obtain \( R_{\text{SVM}} - R_{\text{pGESVM}} = 2.6154 > CD \), \( R_{\text{MCV SVM}} - R_{\text{pGESVM}} = 1.5385 > CD \) and \( R_{\text{GESVM}} - P_{\text{pGESVM}} = 1.3846 > CD \). Thus, the proposed pGESVM algorithm performs significantly better than the rest algorithms.

5. Conclusions

In this paper, a new class of SVMs for multi-class classification has been proposed that exploits geometric data relationships. We have provided direct solutions for the optimization problem solved in both linear and non-linear cases. We have shown that the proposed approach constitutes a general framework for SVM-based classification exploiting geometric data relationships. We have demonstrated the effectiveness of the proposed approach in the human action recognition problem, where state-of-the-art performance has been achieved. We have also applied the proposed classifiers to facial image and standard classification problems where it was shown that they outperform maximum margin classifiers exploiting only intrinsic graph structures.

One limitation of the proposed methods is related to their time complexity. As detailed in Sections 3.1 and 3.2, in order to exploit the geometric data information encoded in the transformed feature space, in the linear case, one should apply a generalized eigenanalysis on an \( D \times D \) matrix, while in the kernel case one should calculate the matrix \( \tilde{K} \) involving the inversion of two \( N \times N \) matrices. Such a process is computationally intensive (especially for the kernel case).
to make the proposed approach applicable in large-scale classification problems, approximate solutions should be devised. We believe that this is an interesting future research direction.

Acknowledgment

This work was supported by the Finnish Funding Agency for Technology and Innovation (TEKES) as part of the Visual label project with A-Lehdet and the Data to Intelligence program of DIGILE (Finnish Strategic Centre for Science, Technology and Innovation in the eld of ICT and digital business).

Bibliography


