

Received September 13, 2020, accepted October 4, 2020, date of publication October 8, 2020, date of current version October 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3029514

Probabilistic Class-Specific Discriminant Analysis

ALEXANDROS IOSIFIDIS¹, (Senior Member, IEEE)

Department of Engineering, Aarhus University, DK-8200 Aarhus, Denmark

e-mail: alexandros.iosifidis@eng.au.dk

This work was supported by the European Union's Horizon 2020 Research and Innovation Programme under Grant 871449 (OpenDR).

ABSTRACT In this paper we formulate a probabilistic model for class-specific discriminant subspace learning. The proposed model can naturally incorporate the multi-modal structure of the negative class, which is neglected by existing class-specific methods. Moreover, it can be directly used to define a class-specific probabilistic classification rule in the discriminant subspace. We show that existing class-specific discriminant analysis methods are special cases of the proposed probabilistic model and, by casting them as probabilistic models, they can be extended to class-specific classifiers. We illustrate the performance of the proposed model in both verification and classification problems.

INDEX TERMS Class-specific discriminant analysis, multi-modal data distributions, verification, classification.

I. INTRODUCTION

Class-Specific Discriminant Analysis (CSDA) [1], [14], [19], [38], determines an optimal subspace suitable for verification problems, where the objective is the discrimination of the class of interest from the rest of the world. As an example, let us consider the person identification problem, either through face verification [22], or through exploiting appearance or movement information [4], [15]. Different from person recognition, which is a multi-class classification problem assigning a sample (facial image or movement sequence) to a class in a pre-defined set of classes (person IDs in this case), person identification discriminates the person of interest from all rest people.

While multi-class discriminant analysis models, like Linear Discriminant Analysis (LDA) and its variants [7], [16], [18], [37], [40], [43] can be applied in such problems, they are inherently limited by the adopted class discrimination definition. That is, the maximal dimensionality of the resulting subspace is restricted by the number of classes, due to the rank of the between-class scatter matrix. In verification problems involving two classes LDA and its variants lead to one-dimensional subspaces. On the other hand, CSDA by expressing class discrimination using the out-of-class and intra-class scatter matrices is able to define subspaces the dimensionality of which is restricted by the the number of samples forming the smallest class (which is usually the

class of interest) or the number of original space dimensions. By defining multiple discriminant dimensions, CSDA has been shown to achieve better class discrimination and better performance in verification problems compared to LDA [1], [19].

While the definition of class discrimination in CSDA and its variants based on the intra-class and out-of-class scatter matrices overcomes the limitations of LDA related to the dimensionality of the discriminant subspace, it overlooks the structure of the negative class. Since in practice samples forming the negative class belong to many classes, different from the positive one, it is expected that they will form subclasses, as illustrated in Figure 1. Class discrimination as defined by CSDA and its variants disregards this structure. Related research in multi-class discriminant analysis indicates that exploitation of subclass information can enhance discrimination power [5], [17], [42], [44].

In this paper, we formulate the class-specific discriminant analysis optimization problem based on a probabilistic model that incorporates the above-described structure of the negative class. We show that the optimization criterion used by standard CSDA and its variants corresponds to a special case of the proposed probabilistic model, while new discriminant subspaces can be obtained by allowing samples of the negative class to form subclasses automatically determined by applying (unsupervised) clustering techniques on the negative class data. Moreover, the use of the proposed probabilistic model for class-specific discriminant learning naturally leads to a classification rule in the discriminant

The associate editor coordinating the review of this manuscript and approving it for publication was Hualong Yu¹.

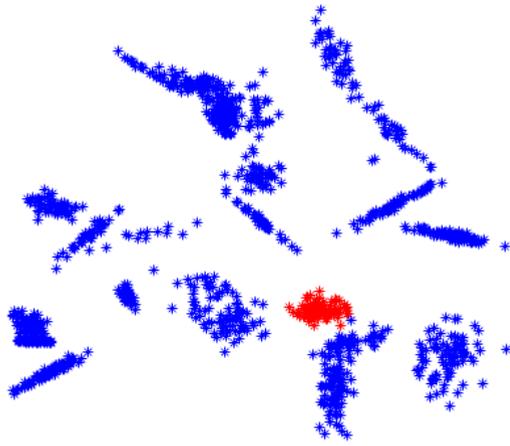


FIGURE 1. A toy example of a 2-dimensional problem illustrating the case where data forming the class of interest (red) is surrounded by data forming other classes. In class-specific learning the labels of the other classes are not available, leading to the assignment of the corresponding data to the negative class label (blue).

subspace, something that is not possible when the standard CSDA criterion is considered.

II. RELATED WORK

Let us denote by $\mathcal{S}_p = \{\mathbf{x}_1, \dots, \mathbf{x}_{N_p}\}$ a set of N_p D -dimensional vectors representing samples of the positive class and by $\mathcal{S}_n = \{\mathbf{x}_{N_p+1}, \dots, \mathbf{x}_N\}$, where $N = N_p + N_n$, a set of N_n vectors representing samples of the negative class. In the following we consider the linear class-specific subspace learning case and we will describe how to perform nonlinear (kernel-based) class-specific subspace learning following the same processing steps in Section III-D. We would like determine a linear projection $\mathbf{W} \in \mathbb{R}^{D \times d}$, mapping \mathbf{x}_i to a d -dimensional subspace, i.e. $\mathbf{z}_i = \mathbf{W}^T \mathbf{x}_i$ that enhances discrimination of the two classes.

Class-specific Discriminant Analysis defines the projection matrix \mathbf{W} as the one maximizing the following criterion:

$$\mathcal{J}(\mathbf{W}) = \frac{\text{Tr}(\mathbf{W}^T \mathbf{S}_n \mathbf{W})}{\text{Tr}(\mathbf{W}^T \mathbf{S}_p \mathbf{W})}, \tag{1}$$

where $\text{Tr}(\cdot)$ is the trace operator. $\mathbf{S}_n \in \mathbb{R}^{D \times D}$ and $\mathbf{S}_p \in \mathbb{R}^{D \times D}$ are the out-of-class and intra-class scatter matrices:

$$\mathbf{S}_n = \sum_{\mathbf{x}_i \in \mathcal{S}_n} (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T \tag{2}$$

$$\mathbf{S}_p = \sum_{\mathbf{x}_i \in \mathcal{S}_p} (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T \tag{3}$$

where \mathbf{m} is the mean vector of the positive class, i.e. $\mathbf{m} = \frac{1}{N_p} \sum_{\mathbf{x}_i \in \mathcal{S}_p} \mathbf{x}_i$. \mathbf{W} is obtained by solving the generalized eigen-analysis problem of $\mathbf{S}_n \mathbf{w} = \lambda \mathbf{S}_p \mathbf{w}$ and keeping the eigenvectors corresponding to the d largest eigenvalues [20]. In the case where \mathbf{S}_n is singular, a regularized version of the above problem is solved.

A Spectral Regression [2] based solution of (1) has been proposed in [1], [19]. Let us denote by \mathbf{w} an eigenvector of

the generalized eigen-analysis problem $\mathbf{S}_n \mathbf{w} = \lambda \mathbf{S}_p \mathbf{w}$ with eigenvalue λ . By setting $\mathbf{X}^T \mathbf{w} = \mathbf{v}$ (\mathbf{X} being the data matrix), the original eigen-analysis problem can be transformed to the following eigen-analysis problem $\mathbf{P}_n \mathbf{v} = \lambda \mathbf{P}_p \mathbf{v}$, where $\mathbf{P}_n = \mathbf{e}_n \mathbf{e}_n^T - \frac{1}{N_p} \mathbf{e}_n \mathbf{e}_p^T - \frac{1}{N_p} \mathbf{e}_p \mathbf{e}_n^T + \frac{1}{N_p^2} \mathbf{e}_p \mathbf{e}_p^T$ and $\mathbf{P}_p = (1 - \frac{2}{N_p} + \frac{1}{N_p^2}) \mathbf{e}_p \mathbf{e}_p^T$. Here $\mathbf{e}_p \in \mathbb{R}^N$ and $\mathbf{e}_n \in \mathbb{R}^N$ are the positive and negative class indicator vectors. \mathbf{W} is then obtained by applying a two-step process:

- Solution of the eigen-analysis problem $\mathbf{P}_n \mathbf{v} = \lambda \mathbf{P}_p \mathbf{v}$ to determine the matrix $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_d]$, where \mathbf{v}_i is the eigenvector corresponding to the i -th largest eigenvalue.
- Calculation of $\mathbf{w}_i, i = 1, \dots, d$ such that $\mathbf{X}^T \mathbf{w}_i = \mathbf{v}_i$.

It has been shown in [12] that the generalized eigenvectors \mathbf{v} of \mathbf{P}_p and \mathbf{P}_n can be directly obtained using the labeling information of the training vectors. However, in that case the order of the eigenvectors is not related to their discrimination ability. It has been also shown in [13] that the class-specific discriminant analysis problem in (1) is equivalent to a low-rank regression problem in which the target vectors are the same as those defined in [12], providing a new proof of the equivalence between class-specific discriminant analysis and class-specific spectral regression.

After determining the data projection matrix \mathbf{W} , the training vectors $\mathbf{x}_i, i = 1, \dots, N$ are mapped to the discriminant subspace $\mathbf{z}_i = \mathbf{W}^T \mathbf{x}_i$. When a classification problem is considered, a classifier is trained using \mathbf{z}_i . For example, [19] trains a linear SVM on vectors $\mathbf{d}_i = |\mathbf{z}_i - \boldsymbol{\mu}|$, where $\boldsymbol{\mu} = \mathbf{W}^T \mathbf{m}$ and the absolute value operator is applied element-wise on the resulting vector. Due to the need of training an additional classifier, class-specific discriminant analysis models are usually employed in class-specific ranking settings, where test vectors \mathbf{x}_j^* are mapped to the discriminant subspace $\mathbf{z}_j^* = \mathbf{W}^T \mathbf{x}_j^*$ and are subsequently ordered based on their distance w.r.t. the positive class mean $d_j = \|\mathbf{z}_j^* - \boldsymbol{\mu}\|_2$.

III. PROBABILISTIC CLASS-SPECIFIC LEARNING

In this section, we follow a probabilistic approach for defining a class-specific discrimination criterion that is able to encode subclass information of the negative class. We call the proposed method Probabilistic Class-Specific Discriminant Analysis (PCSDA). PCSDA defines a subspace $\mathbb{R}^d, d \leq D$ of a feature space \mathbb{R}^D such that the positive class is optimally discriminated from the negative class, based on the assumption that the negative class is formed by multiple subclasses having the same cardinality. We show how to relax these assumptions in the next subsection.

Let us denote by $\mathbf{x} \in \mathbb{R}^D$ a random variable, realizations of which correspond to samples of the positive and negative classes in our problem. PCSDA assumes there exists a positive class c_p following a multivariate Gaussian distribution defined by the mean vector $\mathbf{m} \in \mathbb{R}^D$ and the covariance matrix $\boldsymbol{\Phi}_p \in \mathbb{R}^{D \times D}$, i.e.:

$$P(\mathbf{x}|c_p) \sim N(\mathbf{x}|\mathbf{m}, \boldsymbol{\Phi}_p). \tag{4}$$

Since the negative class c_n is formed by samples belonging to multiple classes (which are not distinguished by the labeling information available during training), PCSDA assumes that the negative class forms subclasses, each of which follows a multivariate Gaussian distribution in \mathbb{R}^D . Let $\mathbf{y} \in \mathbb{R}^D$ be a random variable a realization of which corresponds to the mean vector of a negative subclass. Since in class-specific learning we are interested in maximizing the scatter of the negative data from the positive class (represented by \mathbf{m}), we model the distribution of \mathbf{y} with respect to \mathbf{m} as a multivariate Gaussian distribution:

$$P(\mathbf{y}) \sim N(\mathbf{y}|\mathbf{m}, \Phi_n), \quad (5)$$

where Φ_n is the corresponding covariance matrix expressing the scatter of the negative subclasses with respect to the positive class mean \mathbf{m} . Then, the distribution of the samples from the negative subclasses with respect to the positive class is expressed by the following multi-modal distribution [29]:

$$P(\mathbf{x}|c_n) = \int N(\mathbf{y}|\mathbf{m}, \Phi_n) N(\mathbf{x}|\mathbf{y}, \Phi_w) d\mathbf{y}, \quad (6)$$

where Φ_w is the (common) covariance matrix of the negative subclasses.

A. TRAINING PHASE

Given a set of positive i.i.d. samples $S_p = \{\mathbf{x}_1, \dots, \mathbf{x}_{N_p}\}$ the probability of correct assignment under our model is:

$$P(S_p|c_p) = \prod_{i=1}^{N_p} P(\mathbf{x}_i|c_p). \quad (7)$$

Let us assume that negative class is formed by samples \mathbf{x}_i , $i = N_p + 1, \dots, N$ belonging to K subclasses, i.e. $S_n = \{S_1, \dots, S_K\}$, each having a cardinality of $M = N_n/K$. The probability of assigning each of the negative samples to the corresponding negative subclass is given by:

$$P(S_k|c_n) = \int N(\mathbf{y}|\mathbf{m}, \Phi_n) \prod_{\mathbf{x}_i \in S_k} P(\mathbf{x}_i|\mathbf{y}, \Phi_w) d\mathbf{y}. \quad (8)$$

Without loss of generality, we assume that $\mathbf{m} = \mathbf{0}$ (this can always be done by setting $\mathbf{x}_i - \mathbf{m} \rightarrow \mathbf{x}_i$). Then:

$$P(S_p|c_p) = \frac{1}{(2\pi)^{\frac{N_p D}{2}} |\Phi_p|^{\frac{N_p}{2}}} \exp\left(-\frac{1}{2} Tr(\Phi_p^{-1} S_p)\right) \quad (9)$$

$$P(S_k|c_n) = \frac{1}{M^{-\frac{D}{2}} (2\pi)^{\frac{MD}{2}} |\Phi_w|^{\frac{M-1}{2}} |\Phi_n + \frac{1}{M} \Phi_w|^{\frac{1}{2}}} \cdot \exp\left(-\frac{1}{2} Tr\left(\left(\Phi_n + \frac{1}{M} \Phi_w\right)^{-1} S_n^{(k)}\right)\right) \cdot \exp\left(-\frac{1}{2} Tr(\Phi_w^{-1} S_w^{(k)})\right). \quad (10)$$

In the above, S_p is the scatter matrix of the positive class, i.e. $S_p = \sum_{\mathbf{x}_i \in S_p} \mathbf{x}_i \mathbf{x}_i^T$, $S_w^{(k)}$ is the within-subclass scatter matrix of the k -th negative subclass, i.e. $S_w^{(k)} = \sum_{\mathbf{x}_i \in S_k} (\mathbf{x}_i - \bar{\mathbf{x}}_k)(\mathbf{x}_i - \bar{\mathbf{x}}_k)^T$ and $S_n^{(k)}$ is the scatter matrix of the k -th subclass

w.r.t. the mean vector of the positive class $\mathbf{m} = \mathbf{0}$, i.e. $S_n^{(k)} = \bar{\mathbf{x}}_k \bar{\mathbf{x}}_k^T$, where $\bar{\mathbf{x}}_k$ is the mean vector of the k -th negative subclass, i.e. $\bar{\mathbf{x}}_k = \frac{1}{M} \sum_{\mathbf{x}_i \in S_k} \mathbf{x}_i$. Derivations leading to the above results can be found in the supplementary document. Since the assignment of the negative samples to subclasses is not provided by the labels used during training, we define them by applying a clustering method (e.g. K -Means) on the negative class vectors.

The parameters of the proposed PCSDA are the covariance matrices Φ_p , Φ_n and Φ_w defining the data generation processes for the positive and negative classes. These parameters are estimated by maximizing the (log) probability of correct assignment of the training samples \mathbf{x}_i , $i = 1, \dots, N$:

$$\mathcal{L} = \ln P(S_p|c_p) + \ln P(S_n|c_n), \quad (11)$$

where

$$\ln P(S_p|c_p) = C_1 - \frac{N_p}{2} \ln |\Phi_p| - \frac{1}{2} Tr(\Phi_p^{-1} S_p) \quad (12)$$

and

$$\begin{aligned} \ln P(S_n|c_n) &= \sum_{k=1}^K \ln P(S_k|c_n) = C_2 - \frac{N_n - K}{2} \ln |\Phi_w| \\ &\quad - \frac{K}{2} \ln |\Phi_n + \frac{1}{M} \Phi_w| - \frac{1}{2} Tr(\Phi_w^{-1} S_w) \\ &\quad - \frac{1}{2} Tr\left(\left(\Phi_n + \frac{1}{M} \Phi_w\right)^{-1} S_n\right), \end{aligned} \quad (13)$$

where S_n and S_w are the total scatter matrices of the negative subclasses, i.e., $S_w = \sum_{k=1}^K S_w^{(k)}$ and $S_n = \sum_{k=1}^K S_n^{(k)}$.

By substituting (12) and (13) in (11) the optimization problem takes the form:

$$\begin{aligned} \mathcal{L} &= C_3 - \frac{N_p}{2} \ln |\Phi_p| - \frac{1}{2} Tr(\Phi_p^{-1} S_p) \\ &\quad - \frac{1}{2} Tr(\Phi_w^{-1} S_w) - \frac{1}{2} Tr\left(\left(\Phi_n + \frac{1}{M} \Phi_w\right)^{-1} S_n\right) \\ &\quad - \frac{N_n - K}{2} \ln |\Phi_w| - \frac{K}{2} \ln |\Phi_n + \frac{1}{M} \Phi_w|. \end{aligned} \quad (14)$$

The saddle points of \mathcal{L} with respect to Φ_p , Φ_n , Φ_w lead to:

$$\frac{\partial \mathcal{L}}{\partial \Phi_p} = \mathbf{0} \Rightarrow \Phi_p = \frac{1}{N_p} S_p \quad (15)$$

$$\frac{\partial \mathcal{L}}{\partial \Phi_n} = \mathbf{0} \Rightarrow \Phi_n + \frac{1}{M} \Phi_w = \frac{1}{K} S_n \quad (16)$$

$$\frac{\partial \mathcal{L}}{\partial \Phi_w} = \mathbf{0} \Rightarrow \Phi_w = \frac{1}{N_n - K} S_w. \quad (17)$$

Note that when $N_n = K$ (17) takes an indeterminate form (i.e. $0/0$). However, in this case each negative subclass is formed by one sample and by definition $\Phi_w = \mathbf{0}$. This is discussed more in subsection III-D. Combining (16) and (17) we get:

$$\Phi_n = \frac{1}{K} S_n - \frac{1}{M(N_n - K)} S_w. \quad (18)$$

By combining (17) and (18) we can define the following matrix:

$$\Phi_O = \Phi_n + \Phi_w = \frac{1}{K} S_n + \frac{1}{N_n} S_w, \quad (19)$$

Thus, as can be seen from (15), (17), (18) and (19), the parameters of PCSDA can be calculated using \mathbf{S}_p , \mathbf{S}_n and \mathbf{S}_w defined on the training vectors \mathbf{x}_i , $i = 1, \dots, N$.

Using \mathbf{S}_p , \mathbf{S}_n and \mathbf{S}_w a data transformation $\mathbf{W} \in \mathbb{R}^{D \times D}$ can be obtained by optimizing for:

$$\mathbf{S}_n \mathbf{w} = \lambda (\mathbf{S}_p + \mathbf{S}_w) \mathbf{w}. \quad (20)$$

Problem (20) corresponds to the generalized eigen-analysis problem of the matrices \mathbf{S}_n and $\mathbf{S}_l = \mathbf{S}_p + \mathbf{S}_w$. Since the rank of \mathbf{S}_n is K , the dimensionality of the obtained subspace is restricted to $d \leq \min(D, K)$. Eigenvectors in PCSDA define a data transformation that minimizes the intra-class variance of the positive class and the intra-cluster variance of the negative subclasses, while at the same time maps the means of the negative subclasses as far as possible from the positive class mean. Since the above-described property is optimized by treating (20) as maximization problems, the eigenvectors forming \mathbf{W} are sorted in a decreasing order of the corresponding eigenvalues.

After obtaining \mathbf{W} , the intra-class and out-of-class covariance matrices of the transformed data $\mathbf{z}_i = \mathbf{W}^T \mathbf{x}_i$ are given by:

$$\tilde{\Phi}_p = \mathbf{W}^T \Phi_p \mathbf{W} \quad \text{and} \quad \tilde{\Phi}_O = \mathbf{W}^T \Phi_O \mathbf{W}. \quad (21)$$

The process followed in the training phase of PCSDA is illustrated in Algorithm 1.

In the above we set the assumption that the number of samples forming the negative subclasses is equal. This assumption can be relaxed following one of the following approaches. After assigning all negative samples to subclasses and calculating the negative subclass distributions, one can sample M vectors from each distribution. Alternatively, one can calculate the total within-subclass matrix of the negative class by $\mathbf{S}_w = \sum_{k=1}^K \mathbf{S}_w^{(k)}$. The latter approach is commonly used in multi-class discriminant analysis variants [16]. Note that for the model's parameters calculation, only the overall scatter matrices \mathbf{S}_p , \mathbf{S}_w and \mathbf{S}_n are used.

B. TEST PHASE

After the estimation of the model's parameters, a new sample \mathbf{x}^* can be evaluated. The posterior probabilities of the positive and negative classes are given by:

$$P(c_p | \mathbf{x}^*) = \frac{p(\mathbf{x}^* | c_p) P(c_p)}{p(\mathbf{x}^*)} \quad (22)$$

$$P(c_n | \mathbf{x}^*) = \frac{p(\mathbf{x}^* | c_n) P(c_n)}{p(\mathbf{x}^*)}. \quad (23)$$

The a priori class probabilities $P(c_p)$ and $P(c_n)$ can be calculated by the proportion of the positive and negative samples in the training phase, i.e. $P(c_p) = N_p/N$ and $P(c_n) = N_n/N$. Depending on the problem at hand, it may be sometimes preferable to consider equiprobable classes, i.e. $P(c_p) = P(c_n) = 1/2$, leading to the maximum likelihood classification case. The class-conditional probabilities of the

Algorithm 1 PCSDA: Training Phase

Data: K , d and training data $\{\mathbf{x}_i, l_i\}_{i=1, \dots, N}$

Result: $\tilde{\Phi}_p$, $\tilde{\Phi}_O$, \mathbf{W} , \mathbf{m} , $P(c_p)$, $P(c_n)$

if $K < N_n$ **then**

 | Cluster $\{\mathbf{x}_i\}_{l_i \neq 1}$ to form \mathcal{S}_k , $k = 1, \dots, K$

end if

Calculate \mathbf{m} , \mathbf{S}_p , \mathbf{S}_n and \mathbf{S}_w

Calculate Φ_p from (15) and Φ_O from (19)

Calculate $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_d]$ from (20)

Calculate $\tilde{\Phi}_p$ and $\tilde{\Phi}_O$ from (21)

transformed sample $\mathbf{z}^* = \mathbf{W}^T \mathbf{x}^*$ are given by:

$$\begin{aligned} p(\mathbf{x}^* | c_p) &= \frac{1}{(2\pi)^{\frac{D}{2}} |\Phi_p|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} \mathbf{x}^{*T} \Phi_p^{-1} \mathbf{x}^*\right) \\ &= \frac{1}{(2\pi)^{\frac{D}{2}} |\tilde{\Phi}_p|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} \mathbf{z}^{*T} \tilde{\Phi}_p^{-1} \mathbf{z}^*\right) \end{aligned} \quad (24)$$

and

$$\begin{aligned} p(\mathbf{x}^* | c_n) &= \frac{1}{(2\pi)^{\frac{D}{2}} |\Phi_O|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} \mathbf{x}^{*T} \Phi_O^{-1} \mathbf{x}^*\right) \\ &= \frac{1}{(2\pi)^{\frac{D}{2}} |\tilde{\Phi}_O|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} \mathbf{z}^{*T} \tilde{\Phi}_O^{-1} \mathbf{z}^*\right). \end{aligned} \quad (25)$$

In the above, we used the orthogonal property of the matrix \mathbf{W} . In the case where we want $\mathbf{z}^* \in \mathbb{R}^d$, $d < D$, we keep the dimensions of \mathbf{z}^* corresponding to the first d columns of \mathbf{W} .

Combining (22)-(25) the ratio of class posterior probabilities is equal to:

$$\lambda(\mathbf{z}^*) = \frac{P(c_p) |\tilde{\Phi}_O|^{\frac{1}{2}} \exp\left(-\frac{1}{2} \mathbf{z}^{*T} \tilde{\Phi}_p^{-1} \mathbf{z}^*\right)}{P(c_n) |\tilde{\Phi}_p|^{\frac{1}{2}} \exp\left(-\frac{1}{2} \mathbf{z}^{*T} \tilde{\Phi}_O^{-1} \mathbf{z}^*\right)}. \quad (26)$$

$\lambda(\mathbf{z}^*)$ can be used to classify \mathbf{z}^* to the positive class when $\lambda(\mathbf{z}^*) > 1$ and to the negative class otherwise. Alternatively, (26) can be also used to define the classification rule:

$$\begin{aligned} g(\mathbf{z}^*) &= \ln P(c_p) - \ln P(c_n) \\ &\quad + \frac{1}{2} \ln |\tilde{\Phi}_O| - \frac{1}{2} \ln |\tilde{\Phi}_p| \\ &\quad - \frac{1}{2} \mathbf{z}^{*T} \tilde{\Phi}_p^{-1} \mathbf{z}^* + \frac{1}{2} \mathbf{z}^{*T} \tilde{\Phi}_O^{-1} \mathbf{z}^* \end{aligned} \quad (27)$$

classifying \mathbf{z}^* to the positive class if $g(\mathbf{z}^*) > 0$ and to the negative class otherwise.

In class-specific ranking settings one can follow the process applied when using the standard CSDA approach. First, test vectors \mathbf{x}_j^* are mapped to the discriminant subspace $\mathbf{z}_j^* = \mathbf{W}^T \mathbf{x}_j^*$ and \mathbf{z}_j^* , $i = 1, \dots, N$ are obtained. Then, \mathbf{z}_j^* 's are ordered based on their distance w.r.t. the positive class mean $d_j = \|\mathbf{z}_j^* - \boldsymbol{\mu}\|_2$, where $\boldsymbol{\mu}$ is the mean of positive training samples in \mathbb{R}^d . The process followed in the test phase of PCSDA is illustrated in Algorithm 2.

Algorithm 2 PCSDA: Test Phase

Data: $\tilde{\Phi}_p, \tilde{\Phi}_O, \mathbf{W}, \mathbf{m}, P(c_p), P(c_n)$ and test data $\mathbf{x}_i^*, i = 1, \dots, M$

Result: Predicted labels $l_i, i = 1, \dots, M$ or ranking order $o_i, i = 1, \dots, M$

Calculate $\boldsymbol{\mu} = \mathbf{W}^T \mathbf{m}$ and $\mathbf{z}_i^* = \mathbf{W}^T \mathbf{x}_i^*, i = 1, \dots, M$

if Classification then

for $i = 1:M$ **do**

Calculate $\tilde{\mathbf{z}}_i^* = \mathbf{z}_i^* - \boldsymbol{\mu}$

if $g(\tilde{\mathbf{z}}_i^*) \geq 0$ **then**

$l_i = 1;$

else

$l_i = 0;$

end if

end for

else

Calculate $d_i = \|\boldsymbol{\mu} - \mathbf{z}_i^*\|_2, i = 1, \dots, M$

Sort d_i 's to obtain the order $o_i, i = 1, \dots, M$

end if

C. SPECTRAL REGRESSION SOLUTION OF PCSDA

We further show in the following that PCSDA can be efficiently solved by following a spectral regression based process. Let us denote by $\mathbf{1}_p \in \mathbb{R}^N$ a vector having ones in the elements corresponding to the positive training vectors and zeros elsewhere. In the same manner, we define the vector $\mathbf{1}_k \in \mathbb{R}^N$ as a vector having ones in the elements corresponding to the negative samples belonging to the k -th subclass and zeros elsewhere. We also define by $\mathbf{J}_p \in \mathbb{R}^{N \times N}$ and $\mathbf{J}_k \in \mathbb{R}^{N \times N}$ the diagonal matrices having in their diagonal the vectors $\mathbf{1}_p$ and $\mathbf{1}_k$, respectively. Then, \mathbf{S}_n and \mathbf{S}_l can be expressed as:

$$\mathbf{S}_n = \sum_{k=1}^K (\tilde{\mathbf{x}}_k - \mathbf{m})(\tilde{\mathbf{x}}_k - \mathbf{m})^T = \mathbf{X} \mathbf{L}_n \mathbf{X}^T, \quad (28)$$

$$\mathbf{S}_l = \mathbf{S}_p + \mathbf{S}_w = \mathbf{X} \mathbf{L}_l \mathbf{X}^T \quad (29)$$

where

$$\mathbf{L}_n = \sum_{k=1}^K \left(\frac{1}{N_k^2} \mathbf{1}_k \mathbf{1}_k^T - \frac{1}{N_k N_p} (\mathbf{1}_k \mathbf{1}_p^T + \mathbf{1}_p \mathbf{1}_k^T) + \frac{1}{N_p^2} \mathbf{1}_p \mathbf{1}_p^T \right) \quad (30)$$

$$\mathbf{L}_l = \left(\mathbf{J}_p - \frac{1}{N_p} \mathbf{1}_p \mathbf{1}_p^T \right) + \sum_{k=1}^K \left(\mathbf{J}_k - \frac{1}{N_k} \mathbf{1}_k \mathbf{1}_k^T \right). \quad (31)$$

Substituting (28) and (29) in (20) and setting $\mathbf{v} = \mathbf{X}^T \mathbf{w}$:

$$\mathbf{L}_n \mathbf{v} = \lambda \mathbf{L}_l \mathbf{v}. \quad (32)$$

Thus, the vectors \mathbf{v} are the eigenvectors of the matrix $\mathbf{L} = \tilde{\mathbf{L}}_w^{-1} \mathbf{L}_n$, where $\tilde{\mathbf{L}}_w$ is a regularized version of \mathbf{L}_w , i.e. $\tilde{\mathbf{L}}_w = \mathbf{L}_w + \epsilon \mathbf{I}$ where $\epsilon > 0$. It can be shown that the matrix \mathbf{L} is a block matrix formed by blocks the elements of which are indicated by the class labels of the positive data and the cluster labels of the negative data. The top K eigenvectors

of it are also formed by blocks indicated by the same labels. Thus, they can be defined without the need of solving the problem (32), in a similar manner as in [2], [13]. After the determination of the matrix $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_K]$, the solution of (20) is given by $\mathbf{W} = (\mathbf{X}^T)^\dagger \mathbf{V}$, where $(\mathbf{X}^T)^\dagger$ is the pseudo-inverse of \mathbf{X}^T .

D. CSDA VARIANTS UNDER THE PROBABILISTIC MODEL

A special case of the PCSDA can be obtained by setting the assumption that each negative sample forms a negative subclass, i.e. $K = N_n$ and $M = 1$. In that case $\Phi_w = \mathbf{0}$, $\Phi_O = \Phi_n$, the negative samples are drawn from a distribution $P(\mathbf{x}) \sim N(\cdot | \mathbf{m}, \Phi_n)$, and \mathbf{W} is calculated by solving for $\mathbf{S}_n \mathbf{w} = \tilde{\lambda} \mathbf{S}_p \mathbf{w}$, i.e., we obtain the class discrimination definition of CSDA. The Spectral Regression-based solutions of CSDA in [1], [12], [19] and the low-rank regression solution of [13] use the same class discrimination criterion and, thus, correspond to the same setting of PCSDA. Since the discrimination criterion used in CSDA is a special case of the proposed probabilistic model, all the above-mentioned methods can be extended to perform classification using $g(\cdot)$ in (27). Jointly optimizing the data projection and the classification rule can be considered a big advantage.

E. CLASS-SPECIFIC VS. CONTRASTIVE LEARNING

As can be seen from Eqs. (1), (2) and (3), the Class-specific Discriminant Analysis criterion has similarities with Contrastive Learning based on triplet loss [39], which is defined for multi-class problems. That is, in its generic form, it optimizes for class discrimination by using three samples, i.e. the sample of interest, a sample belonging to the same class with the sample of interest (defining the same-label pair), and a sample belonging to a different class (defining the different-label pair). Thus, it can be modeled by a graph-based criterion, as was shown in [10]. The class-specific learning approaches are defined on binary problems where the negative class is formed data coming from multiple classes, but are assigned the same (negative) label. All the class-specific discriminant analysis approaches (including the proposed one) are defined using variance criteria. They optimize for class discrimination by using the mean of the positive class, a positive sample, and a negative sample. Thus, they can be defined as graph-based methods having the Laplacian matrices \mathbf{L}_n and \mathbf{L}_l in Eqs. (30) and (31).

The adoption of class-specific learning has two advantages compared to contrastive learning when class-specific subspace learning is considered:

- Subspace learning based on triplet loss optimizes for two objectives:
 - 1) Given a positive vector of interest, it should be mapped close to another positive vector (same-label pair), and far away from a negative vector (different-label pair).
 - 2) Given a negative vector of interest, it should be mapped close to another negative vector

(same-label pair), and far away from a positive vector (different-label pair).

While objective 1 is desirable in class-specific problems, optimizing for objective 2 can be destructive. Optimizing for both objectives is good for multi-class problems when assuming class unimodality, like in the case of LDA.

- Class-specific learning defined by a variance-based criterion can lead to fast solutions. This is due to that the graph Laplacian matrices \mathbf{L}_n and \mathbf{L}_l do not need to be calculated and stored in the memory. Instead, their eigenvectors can be efficiently obtained and highly reduce the memory and computational cost. This is described in Section III-C.

F. NON-LINEAR PCSDA

In the above analysis we considered the linear class-specific subspace learning case. In order to non-linearly map $\mathbf{x}_i \in \mathbb{R}^D$ to $\mathbf{z}_i \in \mathbb{R}^d$ traditional kernel-based learning methods perform a non-linear mapping of the input space \mathbb{R}^D to the feature space \mathcal{F} using a function $\phi(\cdot)$, i.e. $\mathbf{x}_i \in \mathbb{R}^D \rightarrow \phi(\mathbf{x}_i) \in \mathcal{F}$. Then, linear class-specific projections are defined by using the training data in \mathcal{F} . Since the dimensionality of \mathcal{F} is arbitrary (virtually infinite), the data representations in \mathcal{F} cannot be calculated. Traditional kernel-based learning methods address this issue by exploiting the Representer theorem and the non-linear mapping is implicitly performed using the kernel function encoding dot products in the feature space, i.e. $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$ [35].

As has been shown in [23] the effective dimensionality of the kernel space \mathcal{F} is at most equal to $L = \min(D, N)$ and, thus, an explicit non-linear mapping $\mathbf{x}_i \in \mathbb{R}^D \rightarrow \Phi_i \in \mathbb{R}^L$ can be calculated such that $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \phi_i^T \phi_j$. This is achieved by using $\Phi = \Sigma^{\frac{1}{2}} \mathbf{U}^T$, where \mathbf{U} and Σ contain the eigenvectors and eigenvalues of the kernel matrix $\mathbf{K} \in \mathbb{R}^{N \times N}$ [24]. Thus, extension of PCSDA to the non-linear (kernel) case can be readily obtained by applying the above-described linear PCSDA on the vectors Φ_i , $i = 1, \dots, N$. Here we need to mention that, since the proposed method exploits kernel-based data representations, it inherits drawbacks of kernel methods related to their high computational and memory costs when applied to large-scale problems. For the cases where the size of training set is prohibitive for applying kernel-based discriminant learning, the Nyström-based kernel subspace learning method of [11] or nonlinear data mappings based on randomized features, as proposed in [34], can be used. We should also note that the application of K -Means using in \mathbb{R}^L corresponds to the application of kernel K -Means in the original space \mathbb{R}^D .

G. TIME COMPLEXITY

The time complexity of PCSDA is as follows [9]:

- K -Means application for determining the subclasses of the negative class having time complexity of $O(\eta KN_n D)$, where η is the number of iterations until convergence,

TABLE 1. Datasets information.

Dataset	D	#Samples	#Classes
Jaffe [27]	1200	210	7
Kanade [21]	1200	245	7
BU [41]	1200	700	7
YALE [26]	1200	2432	38
AR [28]	1200	2600	100
15 scenes [25]	512	4485	15
OptDigits [6]	54	5620	10
Caltech101 [33]	512	9145	102

- Calculation of the scatter matrices in (20) and solution of the generalized eigen-analysis problem having a time complexity of $O(D^3 + D^2 N)$,

Keeping the high-order terms, the overall time complexity is of $O(D^3 + D^2 N)$.

Considering the kernel-based version of PCSDA, its time complexity is as follows:

- Kernel matrix \mathbf{K} calculation having a time complexity of $O(DN^2)$.
- Application of the method [23], involving the eigen-decomposition of \mathbf{K} , having a time complexity of $O(N^3)$.
- K -Means application for determining the negative class subclasses having complexity of the order of $O(\eta KN_n N)$.
- Calculation of the scatter matrices in (20) and solution of the generalized eigen-analysis problem having a time complexity of $O(N^3 + 2N^2)$

Thus, the time complexity of the kernel version of PCSDA is $O(N^3)$, which is the case of a generic kernel-based subspace learning method. As noted in subsection III-F, for applying nonlinear PCSDA on big datasets, the approximate kernel subspace learning method in [11] or nonlinear data mappings based on randomized features, as proposed in [34], can be used leading to $L \ll N$ and highly reducing the overall time complexity to $O(L^3 + LN^2)$.

Finally, considering the Spectral Regression-based solution described in Section III-C, the time complexity of a generic solution is the same as those described above, but efficient solutions are possible by using efficient decomposition methods, like Cholesky decomposition as in [13].

IV. EXPERIMENTS

In this Section we provide experimental results illustrating the performance of the proposed PCSDA method. We used eight datasets in our experiments, details of which are illustrated in Table 1. For BU, Jaffe, Kanade, YALE and AR facial image datasets we used the vectorized pixel intensity values for representing the images. For the 15 scenes and Caltech101 datasets we used deep features generated by average pooling over spatial dimension of the last convolution layer of VGG network [36] trained on ILSVRC2012 database. For optDigits dataset we used the raw features provided by the UCI repository [6].

TABLE 2. Performance on retrieval problems.

	RR	SVR	PCA	LDA	PCSDA-1	PCSDA-K
Jaffe	0.9024 (0.0935)	0.8954 (0.1179)	0.3010 (0.1383)	0.9052 (0.0990)	0.9099 (0.0953)	0.7591 (0.1728)
Kanade	0.6523 (0.2635)	0.6525 (0.2625)	0.2930 (0.1139)	0.6336 (0.2476)	0.6621 (0.2573)	0.4196 (0.2629)
BU	0.6848 (0.2279)	0.6976 (0.2346)	0.2903 (0.1455)	0.6491 (0.2099)	0.6840 (0.1940)	0.6520 (0.1902)
YALE	0.9874 (0.0250)	0.9883 (0.0231)	0.5580 (0.0410)	0.9864 (0.0251)	0.9864 (0.0251)	0.9911 (0.0189)
AR	0.9990 (0.0057)	0.9976 (0.0109)	0.8642 (0.0113)	0.9988 (0.0061)	0.9986 (0.0071)	0.9959 (0.0194)
15 scenes	0.9340 (0.0409)	0.9403 (0.0387)	0.6161 (0.1805)	0.9160 (0.0587)	0.9162 (0.0602)	0.9127 (0.0549)
OptDigits	0.9968 (0.0036)	0.9969 (0.0044)	0.8337 (0.1352)	0.9974 (0.0033)	0.9970 (0.0034)	0.9945 (0.0064)
Caltech101	0.8708 (0.1383)	0.8792 (0.1301)	0.2239 (0.2000)	0.8485 (0.1513)	0.8467 (0.1514)	0.8523 (0.1486)

TABLE 3. Performance on classification problems.

	RR	SVM	PCA+SVM	LDA+NCC	PCSDA-1	PCSDA-K
Jaffe	0.6750 (0.1032)	0.6714 (0.1225)	0.2809 (0.0542)	0.7049 (0.1299)	0.6261 (0.1262)	0.5242 (0.1249)
Kanade	0.4001 (0.1520)	0.4122 (0.1271)	0.2597 (0.0464)	0.5194 (0.2677)	0.4546 (0.2427)	0.3861 (0.1894)
BU	0.4463 (0.0819)	0.4562 (0.1391)	0.2592 (0.0180)	0.5632 (0.1706)	0.5731 (0.1288)	0.5431 (0.1793)
YALE	0.9369 (0.0090)	0.9396 (0.0851)	0.2715 (0.0851)	0.9517 (0.0208)	0.9409 (0.0214)	0.9611 (0.0222)
AR	0.9664 (0.0352)	0.9683 (0.0343)	0.7346 (0.0394)	0.9760 (0.0452)	0.9674 (0.0481)	0.9516 (0.0629)
15 scenes	0.8997 (0.0854)	0.9005 (0.0517)	0.5365 (0.1507)	0.8993 (0.0539)	0.8573 (0.0654)	0.8130 (0.1275)
OptDigits	0.9875 (0.0900)	0.9881 (0.0082)	0.8985 (0.0706)	0.9899 (0.0077)	0.9569 (0.0141)	0.9826 (0.0074)
Caltech101	0.8130 (0.0971)	0.8140 (0.0807)	0.6042 (0.0394)	0.8075 (0.1448)	0.7605 (0.1829)	0.7663 (0.1955)

On each experiment, we formed class-specific ranking and classification problems for each class using the class data as positive samples and the data of the remaining classes as negative samples. In all the experiments the data is non-linearly mapped to the subspace of the kernel space (as discussed in Section III-F). We used the RBF kernel function and set the value of σ equal to the mean pair-wise distance value between the positive training vectors. For the small datasets, we used the method in [24] by keeping the eigenvectors corresponding to all positive eigenvalues, while for the large datasets we used the method in [11] by setting the dimensionality of the resulted kernel subspace to $L = 1000$. On each class-specific problem, we ran five experiments by randomly selecting 70% of the positive and negative classes for training and the rest 30% for testing and measure the average performance value. The hyper-parameter values of all methods have been optimized by applying five-fold cross-validation on the training data. Specifically, for the methods performing subspace learning, the dimensionality of the obtained subspace was selected within the range of [1], [25] and for Support Vector Machine (SVM), Support Vector Regression (SVR) and Ridge Regression (RR) the regularization parameter value was selected in the range of 10^r , $r = -3, \dots, 3$. We used the publicly available software implementations from [8] for the support vector methods. We tested two variants of the proposed PCSDA, one in which the number of negative subclasses is fixed to $K = 1$ (noted as PCSDA-1) and one in which the number of negative subclasses is automatically selected during the training process following the five-fold cross validation process from the set $K = \{5, 10, 15, 20\}$ (noted as PCSDA-K). We opt to automatically determine the number of negative subclasses (and not set K equal to the number of the classes of the multi-class classification problem forming the negative class) to appropriately address

possible issues of class multi-modality. Indeed, by following such an approach, negative subclasses can be formed by samples belonging to more than one classes of the original multi-class classification problem, and multiple negative subclasses can be formed by samples belonging to the same class of the original multi-class classification problem. We should note here that the case $K = 1$ is a subcase of the generic PCSDA-K.

For evaluating the performance of the methods in a ranking setting, we use the mean Average Precision (mAP) metric. For evaluating the performance of the methods in a classification setting, we use the f_1 score defined as $f_1 = 2 \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$. Performance obtained for each method in a ranking and a classification setting is shown in Tables 2 and 3, respectively. The values of the Tables correspond to the mean performance value and the corresponding standard deviation over all class-specific problems of each dataset.

As can be seen in Table 2 for the ranking problems, the use of an unsupervised subspace learning method (PCA) leads to low performance. The use of regression models (RR and SVR) achieves high performance in most of the datasets, except Kanade and BU. The case is similar for the supervised subspace learning methods (LDA and PCSDA). It is interesting to see that in these two datasets PCSDA outperforms LDA by a margin of around 3%, while for the rest datasets their performance is similar. Focusing on PCSDA-1 and PCSDA-K, we can see that in most cases the use of one negative subclass seems to provide (slightly) better results, except for the Yale and Caltech101 datasets. We speculate that this is due to the homogeneity of the remaining datasets, however, we can see that the use of multiple subclasses can enhance performance for heterogeneous data.

Table 3 illustrates the performance of the methods in classification problems. Here we need to note that for performing classification using a subspace-based method (PCA and LDA) one needs to perform a two-step process, i.e. data projection to the subspace followed determined by the method followed by the application of a classification method. We combine PCA with SVM and LDA with Nearest Class Centroid (NCC) classifiers. The choice of LDA+NCC is based on the fact that the LDA optimization problem exploits as a class separability criterion that of NCC. Despite the fact that PCSDA is a subspace learning method, its probabilistic formulation leads to a classification rule (as detailed in Section III-B), and thus there is no need to perform a classification method for PCSDA. We can again see that the use of unsupervised subspace learning combined with SVM leads to low performance. Subspace learning-based classification leads to good performance, outperforming regression-based classification and SVM in most cases. Interestingly, the use of multiple negative subclasses leads to an improvement on Caltech101, Yale and OptDigits datasets of sizes 0.5%, 2% and 2.5%, respectively.

V. CONCLUSIONS

In this paper we proposed a probabilistic model for class-specific discriminant subspace learning that is able to incorporate subclass information of the negative class in the optimization criterion. The adoption of a probability-based optimization criterion for class-specific discriminant subspace learning leads to a classifier defined on the data representations in the discriminant subspace. We showed that the parameters of the probabilistic model can be obtained by applying an efficient Spectral Regression-based process. Moreover, we showed that the proposed probabilistic model includes as special cases existing class-specific discriminant analysis methods. Experimental results illustrated the performance of the proposed model, in comparison with that of related methods, in both verification and classification problems. Interesting extensions of the proposed approach would be to devise class-specific criteria for Subspace Clustering [30], [32] and multi-view based subspace learning [3], [31].

ACKNOWLEDGEMENT

This work has received partial funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant agreement No 871449 (OpenDR). This publication reflects the authors' views only. The European Commission is not responsible for any use that may be made of the information it contains.

REFERENCES

[1] S. R. Arashloo and J. Kittler, "Class-specific kernel fusion of multiple descriptors for face verification using multiscale binarised statistical image features," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 12, pp. 2100–2109, Dec. 2014.

[2] D. Cai, X. He, and J. Han, "Spectral regression for efficient regularized subspace learning," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–7.

[3] G. Cao, A. Iosifidis, K. Chen, and M. Gabbouj, "Generalized multi-view embedding for visual recognition and cross-modal retrieval," *IEEE Trans. Cybern.*, vol. 48, no. 9, pp. 2542–2555, Sep. 2018.

[4] M. Cao, C. Chen, X. Hu, and S. Peng, "Towards fast and kernelized orthogonal discriminant analysis on person re-identification," *Pattern Recognit.*, vol. 94, pp. 218–229, Oct. 2019.

[5] X.-W. Chen and T. Huang, "Facial expression recognition: A clustering-based approach," *Pattern Recognit. Lett.*, vol. 24, nos. 9–10, pp. 1295–1302, Jun. 2003.

[6] D. Dua and C. Graff, "UCI machine learning repository," School Inf. Comput. Sci., Univ. California, Irvine, Irvine, CA, USA, Tech. Rep., 2017. [Online]. Available: https://archive.ics.uci.edu/ml/citation_policy.html

[7] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd ed. Hoboken, NJ, USA: Wiley, 2000.

[8] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, Jun. 2008.

[9] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD, USA: Johns Hopkins Univ. Press, 1996.

[10] L. H. Morsing, O. A. I. Sheikh-Omar, and A. Iosifidis, "Supervised domain adaptation: A graph embedding perspective and a rectified experimental protocol," 2020, *arXiv:2004.11262*. [Online]. Available: <http://arxiv.org/abs/2004.11262>

[11] A. Iosifidis and M. Gabbouj, "Nyström-based approximate kernel subspace learning," *Pattern Recognit.*, vol. 57, pp. 190–197, Sep. 2016.

[12] A. Iosifidis and M. Gabbouj, "Scaling up class-specific kernel discriminant analysis for large-scale face verification," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 11, pp. 2453–2465, Nov. 2016.

[13] A. Iosifidis and M. Gabbouj, "Class-specific kernel discriminant analysis revisited: Further analysis and extensions," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4485–4496, Dec. 2017.

[14] A. Iosifidis, M. Gabbouj, and P. Pekki, "Class-specific nonlinear projections using class-specific kernel spaces," in *Proc. IEEE Int. Conf. Big Data Sci. Eng.*, Aug. 2015, pp. 1–8.

[15] A. Iosifidis, A. Tefas, and I. Pitas, "Activity-based person identification using fuzzy representation and discriminant learning," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 530–542, Apr. 2012.

[16] A. Iosifidis, A. Tefas, and I. Pitas, "On the optimal class representation in linear discriminant analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 9, pp. 1491–1497, Sep. 2013.

[17] A. Iosifidis, A. Tefas, and I. Pitas, "Representative class vector clustering-based discriminant analysis," in *Proc. 9th Int. Conf. Intell. Inf. Hiding Multimedia Signal Process.*, Oct. 2013, pp. 526–529.

[18] A. Iosifidis, A. Tefas, and I. Pitas, "Kernel reference discriminant analysis," *Pattern Recognit. Lett.*, vol. 49, pp. 85–91, Nov. 2014.

[19] A. Iosifidis, A. Tefas, and I. Pitas, "Class-specific reference discriminant analysis with application in human behavior analysis," *IEEE Trans. Human-Machine Syst.*, vol. 45, no. 3, pp. 315–326, Jun. 2015.

[20] Y. Jia, F. Nie, and C. Zhang, "Trace ratio problem revisited," *IEEE Trans. Neural Netw.*, vol. 20, no. 4, pp. 729–735, Apr. 2009.

[21] T. Kanade, J. F. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," in *Proc. 4th IEEE Int. Conf. Automat. Face Gesture Recognit.*, Mar. 2000, pp. 46–53.

[22] Y. P. Kittler, J. Li, and J. Matas, "Face verification using client specific Fisher faces," *Statist. Directions, Shapes Images*, pp. 63–66, 2000.

[23] N. Kwak, "Nonlinear projection trick in kernel methods: An alternative to the kernel trick," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 12, pp. 2113–2119, Dec. 2013.

[24] N. Kwak, "Implementing kernel methods incrementally by incremental nonlinear projection trick," *IEEE Trans. Cybern.*, vol. 47, no. 11, pp. 4003–4009, Nov. 2017.

[25] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2006, pp. 2169–2178.

[26] K.-C. Lee, J. Ho, and D. J. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 684–698, May 2005.

[27] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with Gabor wavelets," in *Proc. 3rd IEEE Int. Conf. Automat. Face Gesture Recognit.*, Apr. 1998, pp. 200–205.

- [28] A. M. Martinez and A. C. Kak, "PCA versus LDA," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 2, pp. 228–233, Feb. 2001.
- [29] A. M. Mood, F. A. Graybill, and D. C. Boes, *Introduction to the Theory of Statistics*, 3rd ed. New York, NY, USA: McGraw-Hill, 1974.
- [30] X. Peng, J. Feng, S. Xiao, W.-Y. Yau, J. T. Zhou, and S. Yang, "Structured autoencoders for subspace clustering," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5075–5086, Oct. 2018.
- [31] X. Peng, Z. Huang, J. Lv, H. Zhu, and J. T. Zhou, "Comic: Multi-view clustering without parameter selection," in *Proc. 36th Int. Conf. Mach. Learn.*, vol. 97, 2019, pp. 5092–5101.
- [32] X. Peng, Z. Yu, Z. Yi, and H. Tang, "Constructing the L2-graph for robust subspace learning and subspace clustering," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 1053–1066, Apr. 2017.
- [33] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 413–420.
- [34] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 1177–1184.
- [35] B. Scholkopf and A. J. Smola, *Learning With Kernels*. Cambridge, MA, USA: MIT Press, 2001.
- [36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: <http://arxiv.org/abs/1409.1556>
- [37] L. S. Souza, B. B. Gatto, J.-H. Xue, and K. Fukui, "Enhanced Grassmann discriminant analysis with randomized time warping for motion recognition," *Pattern Recognit.*, vol. 97, Jan. 2020, Art. no. 107028.
- [38] D. T. Tran, M. Gabbouj, and A. Iosifidis, "Multilinear class-specific discriminant analysis," *Pattern Recognit. Lett.*, vol. 100, pp. 131–136, Dec. 2017.
- [39] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, Feb. 2009.
- [40] H. Ye, Y. Li, C. Chen, and Z. Zhang, "Fast Fisher discriminant analysis with randomized algorithms," *Pattern Recognit.*, vol. 72, pp. 82–92, Dec. 2017.
- [41] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato, "A 3D facial expression database for facial behavior research," in *Proc. 7th Int. Conf. Automat. Face Gesture Recognit. (FGR)*, 2006, pp. 211–216.
- [42] W. Yu and C. Zhao, "Online fault diagnosis in industrial processes using multimodel exponential discriminant analysis algorithm," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 3, pp. 1317–1325, May 2019, doi: 10.1109/TCST.2017.2789188.
- [43] S. Zafeiriou, G. Tzimiropoulos, M. Petrou, and T. Stathaki, "Regularized kernel discriminant analysis with a robust kernel for face recognition and verification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 3, pp. 526–534, Mar. 2012.
- [44] M. Zhu and A. M. Martinez, "Subclass discriminant analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 8, pp. 1274–1286, Aug. 2006.



ALEXANDROS IOSIFIDIS (Senior Member, IEEE) received the B.Sc. degree in electrical and computer engineering and the M.Sc. degree with a specialization in mechatronics from the Democritus University of Thrace, Greece, in 2008 and 2010, respectively, and the Ph.D. degree in computer science from the Aristotle University of Thessaloniki, Greece, in 2014. Before he joined Aarhus University, Denmark, he held Postdoctoral Researcher positions with the Aristotle University of Thessaloniki and Tampere University of Technology, Finland, where he was an Academy of Finland Postdoctoral Research Fellow. He is currently an Associate Professor of Machine Learning with the Department of Engineering, Aarhus University. He has contributed to more than twenty R&D projects financed by EU, Finnish, as well as Danish funding agencies and companies. He has coauthored 75 articles in international journals and 91 papers in international conferences, proposing novel machine learning techniques and their application in a variety of problems. His research interests focus on topics of neural networks and statistical machine learning finding applications in computer vision, financial modeling, and graph mining problems. He has served as an Officer of the Finnish IEEE Signal Processing-Circuits and Systems Chapter (2016–2018). He is currently a member of the EURASIP Technical Area Committee on Visual Information Processing, and serves as an Area/Associate Editor for *Neurocomputing*, *Signal Processing: Image Communications*, *IEEE Access*, and *BMC Bioinformatics journals*.

• • •