Mixture subclass discriminant analysis link to restricted Gaussian model and other generalizations

Nikolaos Gkalelis, Vasileios Mezaris, *Member, IEEE,* Ioannis Kompatsiaris, *Senior Member, IEEE,* and Tania Stathaki

***Abstract*—In this paper, a theoretical link between mixture subclass discriminant analysis (MSDA) and restricted Gaussian model is first presented, and then two further discriminant analysis (DA) methods, fractional step MSDA (FSMSDA) and kernel MSDA (KMSDA) are proposed. Linking MSDA to an appropriate Gaussian model allows the derivation of a new DA method under the Expectation Maximization (EM) framework (EM-MSDA), that derives simultaneously the discriminant subspace as well as the maximum likelihood estimates. The two other proposed methods generalize MSDA in order to solve problems inherited from conventional discriminant analysis. FSMSDA solves the subclass separation problem, that is, the situation when the dimensionality of the discriminant subspace is strictly smaller than the rank of the inter-between-subclass scatter matrix. This is done by an appropriate weighting scheme and the utilization of an iterative algorithm for preserving useful discriminant directions. On the other hand, KMSDA uses the kernel trick to separate data with nonlinearly separable subclass structure. Extensive experimentation shows that the proposed methods outperform conventional MSDA and other LDA variants.**

***Index Terms*—Feature extraction, discriminant analysis, mixture of Gaussians, probabilistic algorithms, clustering, pattern recognition, classification, machine learning.**

I. Introduction

In a natural environment, the high dimensional measurement signals, lying in the F-dimensional measurement space, usually represent patterns residing in a much lower, D-dimensional subspace embedded in the ambient measurement space [1]. Dimensionality reduction (DR) is an important component of statistical pattern classifiers that helps to overcome estimation problems in noisy high-dimensional environments, and thus, often results in improved classifier accuracy as well as lower storage and processing time requirements. A fundamental DR technique is linear discriminant analysis (LDA) [2]–[4]. Given a training set of C classes and N training observations represented with the block matrix $\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_C]$, whose i-th block, $\mathbf{X}_i = [\mathbf{x}_i^1, \ldots, \mathbf{x}_i^{N_i}]$, consists of the N_i observations $\mathbf{x}_i^n \in \mathbb{R}^F$ of the i-th class, this method derives a discriminant subspace spanned by the column vectors of the transformation matrix $\mathbf{\Psi} \in \mathbb{R}^{F \times D}$ that maximizes the ratio

$$J_{LDA}(\mathbf{\Psi}) = \mathrm{tr}(\mathbf{\Psi}^T \mathbf{S}_b \mathbf{\Psi}) / \mathrm{tr}(\mathbf{\Psi}^T \mathbf{S}_w \mathbf{\Psi}) \quad (1)$$

N. Gkalelis is with the Information Technologies Institute/Centre for Research and Technology Hellas (CERTH), Thermi 57001, Greece, and also with the Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, U.K. (email: gkalelis@iti.gr).

V. Mezaris and I. Kompatsiaris are with the Information Technologies Institute/Centre for Research and Technology Hellas (CERTH), Thermi 57001, Greece (email: bmezaris@iti.gr; ikom@iti.gr).

T. Stathaki is with the Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, U.K (email: t.stathaki@imperial.ac.uk).

of the between-class sum of squares $\mathbf{S}_b = \sum_{i=1}^{C} \hat{p}_i (\hat{\boldsymbol{\mu}}_i - \hat{\boldsymbol{\mu}})(\hat{\boldsymbol{\mu}}_i - \hat{\boldsymbol{\mu}})^T$ to the within-class sum of squares $\mathbf{S}_w = \sum_{i=1}^{C} \hat{p}_i \hat{\mathbf{\Sigma}}_i$, where, $\hat{p}_i = N_i/N$, $\hat{\mathbf{\Sigma}}_i = \frac{1}{N_i} \sum_{n=1}^{N_i} (\mathbf{x}_i^n - \hat{\boldsymbol{\mu}}_i)(\mathbf{x}_i^n - \hat{\boldsymbol{\mu}}_i)^T$, $\hat{\boldsymbol{\mu}}_i = \frac{1}{N_i} \sum_{n=1}^{N_i} \mathbf{x}_i^n$, $\hat{\boldsymbol{\mu}} = \sum_{i=1}^{C} \hat{p}_i \hat{\boldsymbol{\mu}}_i$ are the estimated prior, the sample covariance matrix, the sample mean, and the total sample mean, respectively. This optimization problem turns out to be equivalent to the generalized eigenvalue decomposition $\mathbf{S}_b \mathbf{\Psi} = \mathbf{S}_w \mathbf{\Psi} \mathbf{\Lambda}$, where the columns of $\mathbf{\Psi}$ are the generalized eigenvectors corresponding to the largest generalized eigenvalues in the diagonal matrix $\mathbf{\Lambda}$ [5].

Despite its elegant algebraic formulation, two important shortcomings of LDA restrict its use in real-world applications: a) The LDA criterion cannot be applied directly when the matrix $\mathbf{S}_w$ is rank-deficient, a situation that occurs frequently in many applications involving *small sample size* (SSS) data. Several methods have been proposed to deal with this problem, including PCA+LDA [6], MMC LDA [7], dICA [8], and others. b) LDA faces difficulties in deriving a discriminant subspace when the classes are not linearly separable (a problem called hereafter *nonlinearity problem*). This problem has been mostly addressed by using kernel extensions of LDA, [9], [10] or methods that use local linear discriminant analyzers to learn the nonlinear data structure [2], [11]. However, the SSS problem remains, and to address it similar solutions to those discussed above are exploited for both the kernel-based [12], [13] and local-based [14] LDA variants.

Another strategy for solving the nonlinearity problem is to use a clustering procedure to derive a subclass division of the data, and then incorporate this information into the LDA criterion (again, the SSS problem is handled with techniques that overcome the rank-deficiency of $\mathbf{S}_w$, e.g. see [15]). The main advantage of this strategy over the methods described in the previous paragraph (especially over the kernel-based variants of LDA) is that it offers faster computation times during testing, because it only involves a single matrix multiplication. This is the underlying principle of mixture discriminant analysis (MDA) [16] that utilizes the following criterion

$$J_{MDA}(\mathbf{\Psi}) = \mathrm{tr}(\mathbf{\Psi}^T \mathbf{S}_{bs} \mathbf{\Psi}) / \mathrm{tr}(\mathbf{\Psi}^T \mathbf{S}_{ws} \mathbf{\Psi}), \quad (2)$$

where $\mathbf{S}_{bs} = \sum_{i=1}^{C} \sum_{j=1}^{H_i} \hat{p}_{i,j} (\hat{\boldsymbol{\mu}}_{i,j} - \hat{\boldsymbol{\mu}})(\hat{\boldsymbol{\mu}}_{i,j} - \hat{\boldsymbol{\mu}})^T$ is the between-subclass scatter, $\mathbf{S}_{ws} = \sum_{i=1}^{C} \sum_{j=1}^{H_i} \hat{p}_{i,j} \hat{\mathbf{\Sigma}}_{i,j}$ is the within-subclass scatter matrix, H_i denotes the number of subclasses of the i-th class, and $\hat{p}_{i,j}$, $\hat{\boldsymbol{\mu}}_{i,j}$, $\hat{\mathbf{\Sigma}}_{i,j}$ are the estimated prior, sample mean and sample covariance matrix of the j-th subclass of class i.

As our target is to derive a subspace that best separates

observations of different classes, a better choice is to define a discriminant metric that favors the scatter of means between subclasses of different classes. This idea is exploited in subclass discriminant analysis (SDA) [17] that defines the following criterion

$$J_{SDA}(\mathbf{\Psi}) = \mathrm{tr}(\mathbf{\Psi}^T \mathbf{S}_{bsb} \mathbf{\Psi}) / \mathrm{tr}(\mathbf{\Psi}^T \mathbf{\Sigma}_{\mathrm{X}} \mathbf{\Psi}) \,, \qquad (3)$$

where $\mathbf{S}_{bsb} = \sum_{i=1}^{C-1} \sum_{j=1}^{H_i} \sum_{k=i+1}^{C} \sum_{l=1}^{H_k} \hat{p}_{i,j} \hat{p}_{k,l} (\hat{\boldsymbol{\mu}}_{i,j} - \hat{\boldsymbol{\mu}}_{k,l})(\hat{\boldsymbol{\mu}}_{i,j} - \hat{\boldsymbol{\mu}}_{k,l})^T$ is the inter-between-subclass scatter matrix, representing the scatter between the means of subclasses of different classes (inter-subclass scatter of means), and $\mathbf{\Sigma}_{\mathrm{X}} = \frac{1}{N} \sum_{i=1}^{C} \sum_{j=1}^{H_i} \sum_{n=1}^{N_{i,j}} (\mathbf{x}_{i,j}^n - \hat{\boldsymbol{\mu}})(\mathbf{x}_{i,j}^n - \hat{\boldsymbol{\mu}})^T$ is the total covariance matrix.

Several extensions of MDA [18]–[21], and SDA [22]–[25] have been proposed, mainly seeking a more effective subclass partitioning procedure. In [26], mixture subclass discriminant analysis (MSDA) is presented, where it is explained that the use of the criterion

$$J_{MSDA}(\mathbf{\Psi}) = \mathrm{tr}(\mathbf{\Psi}^T \mathbf{S}_{bsb} \mathbf{\Psi}) / \mathrm{tr}(\mathbf{\Psi}^T \breve{\mathbf{\Sigma}}_{\mathrm{X}} \mathbf{\Psi}) \,, \qquad (4)$$

where, $\breve{\mathbf{\Sigma}}_{\mathrm{X}} = \mathbf{S}_{bsb} + \mathbf{S}_{ws}$, is a better choice than the SDA criterion (3). Moreover, this algorithm assumes that the data have a Gaussian homoscedastic subclass structure and introduces an appropriate subclass partitioning procedure along with a nongaussianity criterion to derive the subclass division that optimizes the MSDA criterion. In [26], it was shown that in most cases MSDA outperforms SDA and other LDA variants. However, as we explain in the following, there is still room for further improving dimensionality reduction along the following directions:

1) *Link to Gaussian model*: In [16], [27], [28], it was shown that the LDA and MDA subspaces (defined by the column vectors of the respective projection matrix) coincide with the subspace that maximizes the log-likelihood function of Gaussian class densities or Gaussian mixture class densities, respectively, under the assumption that all class densities (or mixture component densities) are homoscedastic and that all class discriminant information is confined in a D-dimensional subspace of the F-dimensional measurement space. A respective link between MSDA (or SDA) and an appropriate Gaussian model has not yet been provided in the literature, and such a link could lead to a new DR approach.

2) *Subclass separation problem*: When the dimensionality of the LDA subspace is strictly lower than the rank of the between-class matrix, i.e., $D < C - 1$, the projection of the class densities to the discriminant subspace may smear the neighboring classes in the measurement space, a situation described as the class-separation problem [29]–[31]. The same problem can equivalently occur to MSDA (and other subclass variants of LDA), i.e., neighboring subclasses in the original feature space may overlap in the projection subspace when the MSDA subspace dimensionality is strictly lower than the rank of the inter-between-subclass scatter matrix. We refer to this situation as the subclass separation problem.

3) *Subclass nonlinearity problem*: MSDA (and other subclass variants of LDA) can resolve the problem of nonlinearly separable classes as long as a subclass division that results in linearly separable subclasses is identified. If this is not possible, a subclass-based approach that can deal with nonlinearly separable subclasses is desirable, often using an appropriate kernel to map the nonlinearly separable subclass divisions into a new space where they are linearly separable. For instance, in [32], [33] the kernel SDA (KSDA) method was shown to outperform a number of other approaches including kernel discriminant analysis (KDA) [9] and kernel support vector machines (KSVM) [34].

Inspired from the above discussion, in this paper we first provide an explicit link between MSDA and an appropriate Gaussian model, which allows the derivation of a new DA method under the Expectation Maximization framework (EM-MSDA). Furthermore, we present two additional methods, fractional-step MSDA (FSMSDA) and kernel MSDA (KMSDA), to alleviate the subclass separation problem of MSDA and to handle cases where MSDA subclasses are not linearly separable, respectively.

The rest of the paper is structured as follows: In Section II a link between MSDA and a Gaussian model is provided and EM-MSDA is derived, while in Sections III and IV, FSMSDA and KMSDA are presented. In Section V experimental results are reported and Section VI concludes the paper.

II. Link to Gaussian model

In this section, we initially provide a Gaussian mixtures model formulation of the classification task, and then show how the Expectation Maximization (EM) algorithm [35]–[37] can be applied to estimate the unknown model parameters. Through this treatment we provide an explicit link between MSDA and the described Gaussian model, and consequently derive the EM-MSDA algorithm.

A. Gaussian mixtures model

Let $\omega_1, \ldots, \omega_C$ be a finite set of C states of nature (classes) and $(X,\ Y)$ be an $\mathcal{X} \times I_C$-valued random pair, where $\mathcal{X} \subset \mathbb{R}^F$ is the space of observations and $I_C = \{1, \ldots, C\}$ is the class indicator variable [2], [3], [38]. Under this framework we model the i-th class-conditional probability density function $p(\mathbf{x}|\omega_i)$ as a multivariate Gaussian mixture density of H_i component densities where the mixture components along all classes are homoscedastic [16], i.e.

$$p(\mathbf{x}|\omega_i) = \sum_{j=1}^{H_i} \pi_{i,j} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{i,j}) \,, \qquad (5)$$

where, $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{i,j}) = (\tau)^{-F/2} |\mathbf{\Sigma}|^{-1/2} \exp((-1/2)\Delta(\mathbf{x}, \boldsymbol{\mu}_{i,j}))$ is the j-th component density (subclass) of the i-th mixture with constant $\tau \approx 6.283185...$, nonnegative mixing coefficient $\pi_{i,j}$ (satisfying $\sum_{j=1}^{H_i} \pi_{i,j} = 1$), mean vector $\boldsymbol{\mu}_{i,j}$ and covariance matrix $\mathbf{\Sigma}$ shared along all mixture components. Moreover, $\Delta(\mathbf{x}, \boldsymbol{\mu}_{i,j}) = (\mathbf{x} - \boldsymbol{\mu}_{i,j})^T \mathbf{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}_{i,j})$ is the Mahalanobis distance between observation $\mathbf{x}$ and the j-th component of class i.

We then wish to obtain a $D < F$-dimensionality reduction of the data which favors the separability of those subclasses that correspond to different classes. Consequently,

the parameter vector of the presented model is formed as $\boldsymbol{\theta} = [\pi_{1,1}, \boldsymbol{\mu}_{1,1}^T, \ldots, \pi_{C,H_C}, \boldsymbol{\mu}_{C,H_C}^T, \text{vec}(\boldsymbol{\Sigma})^T, \text{vec}(\boldsymbol{\Psi})^T]^T$, where $\boldsymbol{\Psi} \in \mathbb{R}^{F \times D}$ is the required projection matrix for mapping the data into the reduced subspace, T is the vector transposition operator, and the vec() operator stacks the matrix columns to a vector.

B. Log-likelihood function

For the estimation of the unknown parameters $\boldsymbol{\theta}$ we resort to the EM algorithm. The EM algorithm is based on the interpretation of the observed data set $\mathbf{X}_i$ of i-th class as incomplete, where the missing part is a corresponding set $\mathbf{Z}_i = [\mathbf{z}_i^1, \ldots, \mathbf{z}_i^{N_i}]$ of categorical vectors $\mathbf{z}_i^n = [z_{i,1}^n, \ldots, z_{i,H_i}^n]^T$, in which only a particular element $z_{i,\kappa}^n$ equals to 1, indicating that $\mathbf{x}_i^n$ was produced from the κ-th component (subclass) of the i-th mixture density. Under the above formulation and assuming that the C data matrices (blocks) of the block matrix $\mathbf{X}$ (Section I) are independent as well as that the column vectors of the i-th block constitute a random sample from the population with density $p(\mathbf{x}|\omega_i)$ (i.e., all observation vectors are independent and identically distributed (i.i.d.)), the log-likelihood function $\mathcal{L}_1$ of the complete dataset would be (similar to [16], [27] – see Appendix A-A)

$$\begin{aligned} 2\mathcal{L}_1 = & \sum_{i=1}^{C}\sum_{j=1}^{H_i} 2\tilde{N}_{i,j} \ln \pi_{i,j} - NF\ln(2\pi) + N\ln(\det \boldsymbol{\Sigma}^{-1}) \\ & - \sum_{i=1}^{C}\sum_{j=1}^{H_i} \tilde{N}_{i,j}(\bar{\mathbf{x}}_{i,j} - \boldsymbol{\mu}_{i,j})^T \boldsymbol{\Sigma}^{-1}(\bar{\mathbf{x}}_{i,j} - \boldsymbol{\mu}_{i,j}) \\ & - \sum_{i=1}^{C}\sum_{n=1}^{N_i}\sum_{j=1}^{H_i} h_{i,j}^n (\mathbf{x}_i^n - \bar{\mathbf{x}}_{i,j})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i^n - \bar{\mathbf{x}}_{i,j}), \end{aligned} \tag{6}$$

where $h_{i,j}^n$ are the responsibilities, i.e., the expected values of the categorical variables $z_{i,j}^n$ for each data point, given by

$$h_{i,j}^n = \mathbb{E}[z_{i,j}^n] = \frac{\hat{\pi}_{i,j}\mathcal{N}(\mathbf{x}_i^n|\hat{\boldsymbol{\mu}}_{i,j})}{\sum_{j=1}^{H_i}\hat{\pi}_{i,j}\mathcal{N}(\mathbf{x}_i^n|\hat{\boldsymbol{\mu}}_{i,j})}, \tag{7}$$

and $\bar{\mathbf{x}}_{i,j} = (1/\tilde{N}_{i,j})\sum_{n=1}^{N_i} h_{i,j}^n \mathbf{x}_i^n$, $\tilde{N}_{i,j} = \sum_{n=1}^{N_i} h_{i,j}^n$ are the weighted sample mean and the effective number of points of the j-th component of the i-th mixture respectively (note from (7) that $\sum_{j=1}^{H_i}\tilde{N}_{i,j} = N_i$). Moreover, $\ln\delta$ and $\det\mathbf{A}$ denote the natural logarithm of number δ and the determinant of matrix $\mathbf{A}$, respectively. We can rewrite (6) more compactly as

$$\begin{aligned} 2\mathcal{L}_1 &= \zeta - \sum_{i=1}^{C}\sum_{j=1}^{H_i}\tilde{N}_{i,j}(\bar{\mathbf{x}}_{i,j} - \boldsymbol{\mu}_{i,j})^T\boldsymbol{\Sigma}^{-1}(\bar{\mathbf{x}}_{i,j} - \boldsymbol{\mu}_{i,j}) \\ &= \zeta - \text{tr}\{\mathbf{N}(\bar{\mathbf{X}} - \mathbf{M})^T\boldsymbol{\Sigma}^{-1}(\bar{\mathbf{X}} - \mathbf{M})\} \end{aligned} \tag{8}$$

where ζ is the part of the log-likelihood function that is independent of the true means ($\zeta = \sum_{i=1}^{C}\sum_{j=1}^{H_i} 2\tilde{N}_{i,j}\ln\pi_{i,j} - NF\ln(2\pi) + N\ln(\det\boldsymbol{\Sigma}^{-1}) - \sum_{i=1}^{C}\sum_{n=1}^{N_i}\sum_{j=1}^{H_i} h_{i,j}^n(\mathbf{x}_i^n - \bar{\mathbf{x}}_{i,j})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i^n - \bar{\mathbf{x}}_{i,j})$), and $\mathbf{M} = [\boldsymbol{\mu}_{1,1}, \ldots, \boldsymbol{\mu}_{C,H_C}]$, $\bar{\mathbf{X}} = [\bar{\mathbf{x}}_{1,1}, \ldots, \bar{\mathbf{x}}_{C,H_C}]$ are the matrices of true means and weighted sample means respectively.

1) Constrained $\mathbf{M}$*:* We wish to impose two constraints on the values of the true means as we explain in the following. Firstly, we require that the discriminant information is confined in a D-dimensional subspace of the original F-dimensional measurement space (e.g., see p. 339 [27], [16], [28]). Under this restriction the mean of the j-th mixture component of the i-th class density is expressed as

$$\boldsymbol{\mu}_{i,j} = \boldsymbol{\mu}_o + \boldsymbol{\Sigma}\boldsymbol{\Psi}\boldsymbol{\upsilon}_{i,j} \tag{9}$$

where, $\boldsymbol{\Psi} \in \mathbb{R}^{F\times D}$ is a singular transformation matrix with uncorrelated column vectors that transforms $\boldsymbol{\Sigma}$ into the unit matrix

$$\boldsymbol{\Psi}^T\boldsymbol{\Sigma}\boldsymbol{\Psi} = \mathbf{I}, \tag{10}$$

$\boldsymbol{\mu}_o$ is the total mean, and $\boldsymbol{\upsilon}_{i,j} \in \mathbb{R}^D$ is the projection of $\boldsymbol{\mu}_{i,j}$ into the lower-dimensional subspace. The latter is clear if we rearrange (9) to yield $\boldsymbol{\upsilon}_{i,j} = \boldsymbol{\Psi}^T(\boldsymbol{\mu}_{i,j} - \boldsymbol{\mu}_o)$. In matrix form (9) can be written as

$$\mathbf{M} = \mathbf{M}_o + \boldsymbol{\Sigma}\boldsymbol{\Psi}\boldsymbol{\Upsilon} \tag{11}$$

where, $\mathbf{M}$ now is of column rank D, $\mathbf{M}_o = [\boldsymbol{\mu}_o, \ldots, \boldsymbol{\mu}_o]$ is the $F \times H$ matrix whose column vectors equal to the total mean $\boldsymbol{\mu}_o$, and $\boldsymbol{\Upsilon} = [\boldsymbol{\upsilon}_{1,1}, \ldots, \boldsymbol{\upsilon}_{C,H_C}]$ is the matrix with the projection coefficients of the mean vectors.

Secondly, we wish to penalize (6) such that in the lower dimensional subspace the between-subclass spread is emphasized relative to the within-subclass spread. We can impose this by penalizing (6) with the term $\text{tr}\{\boldsymbol{\Upsilon}\mathbf{Q}\boldsymbol{\Upsilon}^T\}$. The penalty matrix $\mathbf{Q}$ is defined as

$$\mathbf{Q} = \mathbf{N}\mathbf{A}^{-1}\mathbf{N} - \mathbf{N} \tag{12}$$

where $\mathbf{N} = \text{diag}(\tilde{N}_{1,1}, \ldots, \tilde{N}_{C,H_C})$ is an $H \times H$ diagonal matrix with diagonal elements the effective sample numbers of the respective mixture component, $\mathbf{A}$ is a symmetric matrix that allows us to express the weighted inter-between-subclass scatter matrix

$$\mathbf{S}_{bsb}^w = \sum_{i=1}^{C-1}\sum_{j=1}^{H_i}\sum_{k=i+1}^{C}\sum_{l=1}^{H_k}\tilde{p}_{i,j}\tilde{p}_{k,l}(\bar{\mathbf{x}}_{i,j} - \bar{\mathbf{x}}_{k,l})(\bar{\mathbf{x}}_{i,j} - \bar{\mathbf{x}}_{k,l})^T, \tag{13}$$

in a matrix product form,

$$\mathbf{S}_{bsb}^w = \bar{\mathbf{X}}\mathbf{A}\bar{\mathbf{X}}^T, \tag{14}$$

and $\bar{\mathbf{X}} = [\bar{\mathbf{x}}_{1,1}, \ldots, \bar{\mathbf{x}}_{C,H_C}]$ is the matrix of the weighted means. That is, the matrix element $\mathbf{A}_{i,j.k,l}$ that corresponds to $\bar{\mathbf{x}}_{i,j}$ and $\bar{\mathbf{x}}_{k,l}$ weighted means takes the value

$$\mathbf{A}_{i,j.k,l} = \begin{cases} \tilde{p}_{i,j}(1-\tilde{p}_i), & \text{if } (i,j) = (k,l), \\ 0 & \text{if } i = k, j \neq l, \\ -\tilde{p}_{i,j}\tilde{p}_{k,l} & \text{else} \end{cases} \tag{15}$$

where $\tilde{p}_{i,j} = \tilde{N}_{i,j}/N$, $\tilde{p}_i = \sum_{j=1}^{H_i}\tilde{p}_{i,j} = N_i/N$. Notice that the sum of the components of any row vector (or any column vector) of matrix $\mathbf{A}$ equals to zero. Therefore, for any matrix with equal column vectors $\mathbf{B} = [\mathbf{b}, \ldots, \mathbf{b}]$ the matrix product $\mathbf{A}\mathbf{B}^T$ will yield the zero matrix. We should also note that $\mathbf{Q}$ is symmetric and that for $\mathbf{A} = \mathbf{N}$, $\mathbf{Q}$ and consequently the penalty term vanish, leading to the conventional MDA algorithm [16]. As we will explain in the

sequel, such a specialization of the penalty matrix $\mathbf{Q}$ will lead to an interesting extension of the MDA algorithm that will provide a subspace equivalent to the MSDA subspace.

2) Constrained $\boldsymbol{\Sigma}$: Similarly to MDA with centroid shrinking (p.171, [16]) we constrain the covariance matrix at the weighted within-subclass scatter matrix

$$\begin{aligned}\hat{\boldsymbol{\Sigma}} = \mathbf{S}_{ws}^{w} &= \sum_{i=1}^{C}\sum_{j=1}^{H_i} \tilde{p}_{i,j}\hat{\boldsymbol{\Sigma}}_{i,j}^{w} \\ &= \frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{H_i}\sum_{n=1}^{N_i} h_{i,j}^{n}(\mathbf{x}_i^n - \bar{\mathbf{x}}_{i,j})(\mathbf{x}_i^n - \bar{\mathbf{x}}_{i,j})^T ,\end{aligned} \tag{16}$$

where $\hat{\boldsymbol{\Sigma}}_{i,j}^{w} = (1/\tilde{N}_{i,j})\sum_{n=1}^{N_i} h_{i,j}^{n}(\mathbf{x}_i^n - \bar{\mathbf{x}}_{i,j})(\mathbf{x}_i^n - \bar{\mathbf{x}}_{i,j})^T$ is the weighted sample covariance matrix of (i,j) component density.

Imposing the above constraints (10), (11), (16), and the penalty term (12) in (8), we finally arrive to the following penalized and restricted version of the log-likelihood function

$$\begin{aligned}2\mathcal{L}_2 = &-\operatorname{tr}\{\mathbf{N}(\bar{\mathbf{X}} - \mathbf{M}_o - \hat{\boldsymbol{\Sigma}}\boldsymbol{\Psi}\boldsymbol{\Upsilon})^T\hat{\boldsymbol{\Sigma}}^{-1}(\bar{\mathbf{X}} - \mathbf{M}_o - \hat{\boldsymbol{\Sigma}}\boldsymbol{\Psi}\boldsymbol{\Upsilon})\} \\ &- \operatorname{tr}\{\boldsymbol{\Upsilon}\mathbf{Q}\boldsymbol{\Upsilon}^T\} + \zeta,\end{aligned} \tag{17}$$

where $\boldsymbol{\Psi}$ is constrained by (10).

C. EM algorithm

The EM algorithm can be applied to obtain the maximum likelihood estimate (MLE) of the model parameters in (17). This algorithm alternates between two steps, the Expectation step (E-step) and the Maximization step (M-step), to produce a sequence of estimates until some convergence criterion is met.

1) E-step: During the E-step, the parameter values identified in the previous EM cycle are used to compute the responsibilities $h_{i,j}^n$ using (7).

2) M-step: In this step, the unknown mixture parameters are estimated by maximizing (17). In particular, we need to estimate the mixing coefficients $\pi_{i,j}$ and the true means $\boldsymbol{\mu}_{i,j}$ for each mixture component in (5).

Estimation of $\pi_{i,j}$: The mixing coefficients are estimated by maximizing (17) subject to the constraint that $\sum_{j=1}^{H_i}\pi_{i,j} = 1$, giving (similarly to [37] – see Appendix A-B)

$$\hat{\pi}_{i,j} = \frac{\tilde{N}_{i,j}}{N_i}\,. \tag{18}$$

Estimation of $\boldsymbol{\mu}_{i,j}$: Now we proceed to estimating the true means in $\mathbf{M}$, or equivalently $\mathbf{M}_o$, $\boldsymbol{\Upsilon}$, and $\boldsymbol{\Psi}$, that maximize (17) subject to $\boldsymbol{\Psi}^T\hat{\boldsymbol{\Sigma}}\boldsymbol{\Psi} = \mathbf{I}$. In (17) ζ is independent of $\mathbf{M}$ and thus can be discarded from the optimization criterion. Moreover, the maximization of $\mathcal{L}_2$ is equivalent to the minimization of $-\mathcal{L}_2$ under the same conditions, leading us to the following optimization problem

$$\underset{\mathbf{M}_o,\boldsymbol{\Upsilon},\boldsymbol{\Psi}}{\operatorname{argmin}}\ \mathcal{L}_3 \ \text{ subject to } \ \boldsymbol{\Psi}^T\hat{\boldsymbol{\Sigma}}\boldsymbol{\Psi} = \mathbf{I}, \tag{19}$$

where,

$$\begin{aligned}\mathcal{L}_3 = &\operatorname{tr}\{\mathbf{N}(\bar{\mathbf{X}} - \mathbf{M}_o - \hat{\boldsymbol{\Sigma}}\boldsymbol{\Psi}\boldsymbol{\Upsilon})^T\hat{\boldsymbol{\Sigma}}^{-1}(\mathbf{X} - \mathbf{M}_o - \hat{\boldsymbol{\Sigma}}\boldsymbol{\Psi}\boldsymbol{\Upsilon})\} \\ &+ \operatorname{tr}\{\boldsymbol{\Upsilon}\mathbf{Q}\boldsymbol{\Upsilon}^T\}.\end{aligned} \tag{20}$$

Setting $\bar{\mathbf{y}}_{i,j} = \hat{\boldsymbol{\Sigma}}^{-1/2}\bar{\mathbf{x}}_{i,j}$, $\mathbf{v}_{i,j} = \hat{\boldsymbol{\Sigma}}^{-1/2}\boldsymbol{\mu}_{i,j}$, and $\mathbf{v}_o = \hat{\boldsymbol{\Sigma}}^{-1/2}\boldsymbol{\mu}_o$, we can write $\mathbf{v}_{i,j} = \mathbf{v}_o + \tilde{\boldsymbol{\Psi}}\boldsymbol{\upsilon}_{i,j}$ or in matrix form

$$\mathbf{V} = \mathbf{V}_o + \tilde{\boldsymbol{\Psi}}\boldsymbol{\Upsilon} \tag{21}$$

where $\tilde{\boldsymbol{\Psi}} = \hat{\boldsymbol{\Sigma}}^{1/2}\boldsymbol{\Psi}$, $\bar{\mathbf{Y}} = \hat{\boldsymbol{\Sigma}}^{-1/2}\bar{\mathbf{X}} = [\bar{\mathbf{y}}_{1,1},\ldots,\bar{\mathbf{y}}_{C,H_C}]$, $\mathbf{V} = \hat{\boldsymbol{\Sigma}}^{-1/2}\mathbf{M} = [\mathbf{v}_{1,1},\ldots,\mathbf{v}_{C,H_C}]$, and $\mathbf{V}_o = \hat{\boldsymbol{\Sigma}}^{-1/2}\mathbf{M}_o$. Substituting this to (20) we arrive to

$$\begin{aligned}\mathcal{L}_3 &= \sum_{i=1}^{C}\sum_{j=1}^{H_i}\tilde{N}_{i,j}(\bar{\mathbf{y}}_{i,j} - \mathbf{v}_{i,j})^T(\bar{\mathbf{y}}_{i,j} - \mathbf{v}_{i,j}) \\ &= \operatorname{tr}\{\mathbf{N}(\bar{\mathbf{Y}} - \mathbf{V})^T(\bar{\mathbf{Y}} - \mathbf{V})\} + \operatorname{tr}\{\boldsymbol{\Upsilon}\mathbf{Q}\boldsymbol{\Upsilon}^T\} \\ &= \operatorname{tr}\{\mathbf{N}(\bar{\mathbf{Y}} - \mathbf{V}_o - \tilde{\boldsymbol{\Psi}}\boldsymbol{\Upsilon})^T(\bar{\mathbf{Y}} - \mathbf{V}_o - \tilde{\boldsymbol{\Psi}}\boldsymbol{\Upsilon})\} \\ &\quad + \operatorname{tr}\{\boldsymbol{\Upsilon}\mathbf{Q}\boldsymbol{\Upsilon}^T\}\end{aligned} \tag{22}$$

Setting the derivatives of $\mathcal{L}_3$ in (22) with respect to the projection coefficients $\boldsymbol{\Upsilon}$ to zero we obtain

$$\frac{\partial\mathcal{L}_3}{\partial\boldsymbol{\Upsilon}} = 0 \Rightarrow \boldsymbol{\Upsilon} = \tilde{\boldsymbol{\Psi}}^T(\bar{\mathbf{Y}} - \mathbf{V}_o)\mathbf{A}\mathbf{N}^{-1} \tag{23}$$

We can now expand (22) as

$$\begin{aligned}\mathcal{L}_3 &= \operatorname{tr}\{\mathbf{N}(\bar{\mathbf{Y}} - \mathbf{Y}_o + \mathbf{Y}_o - \mathbf{V}_o - \tilde{\boldsymbol{\Psi}}\boldsymbol{\Upsilon})^T \\ &\quad \times(\bar{\mathbf{Y}} - \mathbf{Y}_o + \mathbf{Y}_o - \mathbf{V}_o - \tilde{\boldsymbol{\Psi}}\boldsymbol{\Upsilon})\} + \operatorname{tr}\{\boldsymbol{\Upsilon}\mathbf{Q}\boldsymbol{\Upsilon}^T\} \\ &= \operatorname{tr}\{\mathbf{N}(\bar{\mathbf{Y}} - \mathbf{Y}_o)^T(\bar{\mathbf{Y}} - \mathbf{Y}_o)\} \\ &\quad + \operatorname{tr}\{\mathbf{N}(\mathbf{Y}_o - \mathbf{V}_o)^T(\mathbf{Y}_o - \mathbf{V}_o)\} \\ &\quad + \operatorname{tr}\{\mathbf{N}\boldsymbol{\Upsilon}^T\tilde{\boldsymbol{\Psi}}^T\tilde{\boldsymbol{\Psi}}\boldsymbol{\Upsilon}\} + \operatorname{tr}\{\boldsymbol{\Upsilon}\mathbf{Q}\boldsymbol{\Upsilon}^T\} \\ &\quad + 2\operatorname{tr}\{\mathbf{N}(\bar{\mathbf{Y}} - \mathbf{Y}_o)^T(\mathbf{Y}_o - \mathbf{V}_o)\} \\ &\quad - 2\operatorname{tr}\{\mathbf{N}(\bar{\mathbf{Y}} - \mathbf{V}_o)^T\tilde{\boldsymbol{\Psi}}\boldsymbol{\Upsilon}\}\end{aligned} \tag{24}$$

Reformulating the fifth term of (24) we see that it vanishes

$$\begin{aligned}&\operatorname{tr}\{\mathbf{N}(\bar{\mathbf{Y}} - \mathbf{Y}_o)^T(\mathbf{Y}_o - \mathbf{V}_o)\} \\ &\quad = \textstyle\sum_{i=1}^{C}\sum_{j=1}^{H}\tilde{N}_{i,j}(\bar{\mathbf{y}}_{i,j} - \mathbf{y}_o)^T(\mathbf{y}_o - \mathbf{v}_o) = 0\,,\end{aligned} \tag{25}$$

Using (12), (23) and taking into account that $\tilde{\boldsymbol{\Psi}}^T\tilde{\boldsymbol{\Psi}} = \boldsymbol{\Psi}^T\hat{\boldsymbol{\Sigma}}\boldsymbol{\Psi} = \mathbf{I}$ the summand of the third and forth term of (24) becomes

$$\begin{aligned}&\operatorname{tr}\{\mathbf{N}\boldsymbol{\Upsilon}^T\tilde{\boldsymbol{\Psi}}^T\tilde{\boldsymbol{\Psi}}\boldsymbol{\Upsilon}\} + \operatorname{tr}\{\boldsymbol{\Upsilon}\mathbf{Q}\boldsymbol{\Upsilon}^T\} \\ &\quad = \operatorname{tr}\{\boldsymbol{\Upsilon}(\mathbf{N} + \mathbf{Q})\boldsymbol{\Upsilon}^T\} = \operatorname{tr}\{\boldsymbol{\Upsilon}\mathbf{N}\mathbf{A}^{-1}\mathbf{N}\boldsymbol{\Upsilon}^T\} \\ &\quad = \operatorname{tr}\{\mathbf{A}(\bar{\mathbf{Y}} - \mathbf{V}_o)^T\tilde{\boldsymbol{\Psi}}\tilde{\boldsymbol{\Psi}}^T(\bar{\mathbf{Y}} - \mathbf{V}_o)\}\end{aligned} \tag{26}$$

and similarly using (23) the sixth term of (24) becomes

$$\begin{aligned}&\operatorname{tr}\{\mathbf{N}(\bar{\mathbf{Y}} - \mathbf{V}_o)^T\tilde{\boldsymbol{\Psi}}\boldsymbol{\Upsilon}\} \\ &\quad = \operatorname{tr}\{\mathbf{A}(\bar{\mathbf{Y}} - \mathbf{V}_o)^T\tilde{\boldsymbol{\Psi}}\tilde{\boldsymbol{\Psi}}^T(\bar{\mathbf{Y}} - \mathbf{V}_o)\}\end{aligned} \tag{27}$$

Substituting (25), (26), (27) into (24) we arrive to

$$\begin{aligned}\mathcal{L}_3 &= \operatorname{tr}\{\mathbf{N}(\bar{\mathbf{Y}} - \mathbf{Y}_o)^T(\bar{\mathbf{Y}} - \mathbf{Y}_o)\} \\ &\quad + \operatorname{tr}\{\mathbf{N}(\mathbf{Y}_o - \mathbf{V}_o)^T(\mathbf{Y}_o - \mathbf{V}_o)\} \\ &\quad - \operatorname{tr}\{\mathbf{A}(\bar{\mathbf{Y}} - \mathbf{V}_o)^T\tilde{\boldsymbol{\Psi}}\tilde{\boldsymbol{\Psi}}^T(\bar{\mathbf{Y}} - \mathbf{V}_o)\}\end{aligned} \tag{28}$$

Using the fact that $\mathbf{A}\mathbf{V}_o^T = \mathbf{0}$, the last term of (28) is simplified to

$$\begin{aligned}&\operatorname{tr}\{\mathbf{A}(\bar{\mathbf{Y}} - \mathbf{V}_o)^T\tilde{\boldsymbol{\Psi}}\tilde{\boldsymbol{\Psi}}^T(\bar{\mathbf{Y}} - \mathbf{V}_o)\} = \operatorname{tr}\{\mathbf{A}\bar{\mathbf{Y}}^T\tilde{\boldsymbol{\Psi}}\tilde{\boldsymbol{\Psi}}^T\bar{\mathbf{Y}}\} \\ &\quad + \operatorname{tr}\{\mathbf{A}\mathbf{V}_o^T\tilde{\boldsymbol{\Psi}}\tilde{\boldsymbol{\Psi}}^T\mathbf{V}_o\} - 2\operatorname{tr}\{\mathbf{A}\mathbf{V}_o^T\tilde{\boldsymbol{\Psi}}\tilde{\boldsymbol{\Psi}}^T\bar{\mathbf{Y}}\} \\ &\quad = \operatorname{tr}\{\mathbf{A}\bar{\mathbf{Y}}^T\tilde{\boldsymbol{\Psi}}\tilde{\boldsymbol{\Psi}}^T\bar{\mathbf{Y}}\}\,,\end{aligned} \tag{29}$$

and substituting this back to (28) we arrive to

$$\mathcal{L}_3 = \mathrm{tr}\{\mathbf{N}(\bar{\mathbf{Y}}-\mathbf{Y}_o)^T(\bar{\mathbf{Y}}-\mathbf{Y}_o)\} + \mathrm{tr}\{\mathbf{N}(\mathbf{Y}_o-\mathbf{V}_o)^T(\mathbf{Y}_o-\mathbf{V}_o)\} - \mathrm{tr}\{\tilde{\mathbf{\Psi}}^T\bar{\mathbf{Y}}\mathbf{A}\bar{\mathbf{Y}}^T\tilde{\mathbf{\Psi}}\}, \quad (30)$$

where, $\mathbf{Y}_o = \hat{\mathbf{\Sigma}}^{-1/2}\mathbf{X}_o$, and $\mathbf{X}_o$ is the $F \times H$ matrix whose column vectors equal to the weighted mean $\mathbf{x}_o = (1/N)\sum_{i=1}^{C}\sum_{j=1}^{H_i}\sum_{n=1}^{N_i} h_{i,j}^n \mathbf{x}_i^n = \sum_{i=1}^{C}\sum_{j=1}^{H_i}\tilde{p}_{i,j}\bar{\mathbf{x}}_{i,j}$. We now have to minimize (30) with respect to $\mathbf{V}_o$, or equivalently

$$\mathrm{tr}\{\mathbf{N}(\mathbf{Y}_o-\mathbf{V}_o)^T(\mathbf{Y}_o-\mathbf{V}_o)\} = N(\mathbf{y}_o-\mathbf{v}_o)^T(\mathbf{y}_o-\mathbf{v}_o) \quad (31)$$

which is minimized for $\mathbf{y}_o = \mathbf{v}_o$ and, thus, yielding $\hat{\boldsymbol{\mu}}_o = \mathbf{x}_o$ or in matrix form

$$\hat{\mathbf{M}}_o = \mathbf{X}_o\,. \quad (32)$$

Without loss of generality we can set $\mathbf{x}_o = [0,\ldots,0]^T$ (e.g. setting $\mathbf{X} \leftarrow \mathbf{X}-\mathbf{X}_o$). Substituting this back to (30) we arrive to

$$\mathcal{L}_3 = \mathrm{tr}\{\hat{\mathbf{\Sigma}}^{-1}\bar{\mathbf{X}}\mathbf{N}\bar{\mathbf{X}}^T\} - \mathrm{tr}\{\mathbf{\Psi}^T\bar{\mathbf{X}}\mathbf{A}\bar{\mathbf{X}}^T\mathbf{\Psi}\} \quad (33)$$

where we have used the requirement that $\mathbf{\Psi}$ transforms the pooled covariance matrix $\hat{\mathbf{\Sigma}}$ into the unit matrix ($\mathbf{\Psi}^T\hat{\mathbf{\Sigma}}\mathbf{\Psi} = \mathbf{I}$). In (33) only the second term depends on the transformation matrix, and, thus, this matrix can be obtained by solving the following optimization problem

$$\underset{\mathbf{\Psi}}{\mathrm{argmax}}\ \mathrm{tr}\{\mathbf{\Psi}^T\mathbf{S}_{bsb}^w\mathbf{\Psi}\} \quad \text{subject to} \quad \mathbf{\Psi}^T\mathbf{S}_{ws}^w\mathbf{\Psi} = \mathbf{I} \quad (34)$$

where we have used (14) and fixed $\hat{\mathbf{\Sigma}}$ according to (16). The solution to this problem is obtained by the set $\{\boldsymbol{\psi}_i | i = 1,\ldots,D\}$ of the generalized eigenvectors of $\mathbf{S}_{bsb}^w$ and $\mathbf{S}_{ws}^w$ corresponding to the D largest eigenvalues $\{\lambda_i | i = 1,\ldots,D\}$ of the following generalized eigenvalue decomposition [2]

$$\mathbf{S}_{bsb}^w\mathbf{\Psi} = \mathbf{S}_{ws}^w\mathbf{\Psi}\mathbf{\Lambda} \quad (35)$$

where $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1,\ldots,\lambda_D)$. Therefore, the subspace that maximizes the constrained log-likelihood function in (19) at each EM cycle coincides with the subspace that maximizes the MSDA criterion, where the scatter matrices in (4) are replaced by their weighted equivalent in each EM cycle. The MLE of the true means can now be computed by substituting (23), (32) into (11) and using the computed estimates of (16), (35) for $\mathbf{S}_{ws}$ and $\mathbf{\Psi}$ respectively

$$\begin{aligned}\hat{\mathbf{M}} &= \mathbf{X}_o + \mathbf{S}_{ws}^w\mathbf{\Psi}\mathbf{\Psi}^T(\bar{\mathbf{X}}-\mathbf{X}_o)\mathbf{A}\mathbf{N}^{-1} \\ &= \mathbf{S}_{ws}^w\mathbf{\Psi}\mathbf{\Psi}^T\bar{\mathbf{X}}\mathbf{A}\mathbf{N}^{-1} \end{aligned} \quad (36)$$

where, we have assumed that $\mathbf{X}_o = \mathbf{0}$.

D. Model selection

The Gaussian model described above as well as the derived EM algorithm assume that the number of mixing components in each Gaussian mixture density is provided. However, this information is rarely known. In order to estimate the optimum number of mixing components for each mixture density with respect to the given training set, we utilize an iterative procedure, where at each iteration a new Gaussian model is specified (with respect to the number of mixture components) and a nongaussianity measure Φ is evaluated in order to assess the goodness of fit of the particular Gaussian model. This iterative process is repeated until the nongaussianity measure Φ converges to a small value as explained in the following.

Skewness and kurtosis can be used to provide an indication of how well a particular Gaussian mixture density fits the training data of a specific class [26], [39], [40]. Estimates of the weighted standardized skewness $\hat{\beta}_{i,j,f}$ and kurtosis $\hat{\gamma}_{i,j,f}$ along the f-th dimension regarding the j-th mixture component of the i-th class can be computed as follows

$$\hat{\beta}_{i,j,f} = \frac{\frac{1}{\bar{N}_{i,j}}\sum_{n=1}^{N_i} h_{i,j}^n (x_{i,j,f}^n - \hat{\mu}_{i,j,f})^3}{\hat{\sigma}_{i,j,f}^3}\,, \quad (37)$$

$$\hat{\gamma}_{i,j,f} = \frac{\frac{1}{\bar{N}_{i,j}}\sum_{n=1}^{N_i} h_{i,j}^n (x_{i,j,f}^n - \hat{\mu}_{i,j,f})^4}{\hat{\sigma}_{i,j,f}^4} - 3\,, \quad (38)$$

where $x_{i,j,f}^n$ is the f-th element of $\mathbf{x}_{i,j}^n$, and $\hat{\mu}_{i,j,f}, \hat{\sigma}_{i,j,f}$ are the sample mean and standard deviation of the j-th mixture of i-th class along the f-th dimension. The above estimates will be close to zero for Gaussian densities and deviate from zero the more the underlying density deviates from the Gaussian. We can thus obtain an estimate of the skewness $\hat{\beta}_{i,j}$ and kurtosis $\hat{\gamma}_{i,j}$ of the (i,j) component density by averaging along all dimensions, i.e., $\hat{\beta}_{i,j} = (1/F)\sum_{f=1}^{F}|\hat{\beta}_{i,j,f}|$, $\hat{\gamma}_{i,j} = (1/F)\sum_{f=1}^{F}|\hat{\gamma}_{i,j,f}|$, where $|a|$ denotes absolute value of a. Similarly, we can define a nongaussianity measure regarding the Gaussian mixture density referring to the i-th class using

$$\Phi_i = \sum_{j=1}^{H_i}\hat{\pi}_{i,j}(\hat{\beta}_{i,j} + \hat{\gamma}_{i,j})\,. \quad (39)$$

A large value of Φ_i will denote that the respective Gaussian mixture density does not fit well the underlying density function of the i-th class training data. Therefore, at each iteration this measure is used to select the mixture density that yielded the worst fit according to the following criterion

$$k = \underset{i=1,\ldots,C}{\mathrm{argmax}}(\Phi_i)\,, \quad (40)$$

and the required number of mixture components referring to this mixture density is increased by one ($H_k \leftarrow H_k + 1$). Similarly, at each iteration a total nongaussianity measure is defined for assessing the fitness of the current Gaussian model with respect to the overall training data set

$$\Phi = \sum_{i=1}^{C}\tilde{p}_i\Phi_i\,. \quad (41)$$

The value of Φ is examined at each iteration, and the iterative procedure is completed upon the convergence of Φ to a steady-state solution. The resulting EM-MSDA algorithm is outlined in Algorithm 1. Alternatively, a cross-validation criterion can be used to select the Gaussian model that provides the best empirical recognition rate.

III. Fractional step mixture subclass discriminant analysis

In equivalence to the class separation problem of LDA [29]–[31], the subclass separation problem may occur when the dimensionality of the MSDA subspace D is strictly lower

Algorithm 1 EM-MSDA

Input: Annotated data set $\mathbf{X}$
Output: $\mathbf{\Psi}$

Initialize: $H_1 = \cdots = H_C = 1$, $H = C$, Φ_i (39), Φ (41)
repeat
  Compute class label k of class to repartition (40)
  Set: $H_k \leftarrow H_k + 1$
  Repartition k-th class to H_k subclasses using k-means
  Initialize the MLE parameters $\hat{\boldsymbol{\theta}}$
  repeat
    E-step: Compute responsibilities $h_{i,j}^n$ (7)
    M-step: Compute MLEs: $\mathbf{\Psi}$ (35), $\hat{\boldsymbol{\theta}}$ (16), (36)
  until convergence of $\hat{\boldsymbol{\theta}}$
  Compute nongaussianity Φ_i for each class (39)
  Compute total nongaussianity Φ (41)
until convergence of Φ

than the rank of the inter-between-subclass scatter matrix ($D < \text{rank}(\mathbf{S}_{bsb}) \leq \min(F, H-1)$). When this happens, distinct subclasses in the measurement space may not separate well in the lower dimensional subspace. To demonstrate this problem we use an artificial dataset of two classes, where, the first class consists of two Gaussian subclasses $\mathcal{N}_{1,1}$, $\mathcal{N}_{1,2}$, and the second class is a unimodal Gaussian $\mathcal{N}_{2,1}$. The means of the Gaussian distributions are $\boldsymbol{\mu}_{1,1} = [6\ 22]^T$, $\boldsymbol{\mu}_{1,2} = [0\ 0]^T$, $\boldsymbol{\mu}_{2,1} = [12\ 22]^T$, whereas a common covariance matrix is shared along all distributions $\mathbf{\Sigma} = [0.7\ 0.3\ ;\ 0.3\ 0.7]$, as depicted in Figure 1. Under these settings, we see that the one-dimensional projection transformation derived using MSDA ($\boldsymbol{\psi}_{MSDA}$) causes a large overlap between the subclasses $\mathcal{N}_{1,1}$ and $\mathcal{N}_{2,1}$, which are close to each other, but well separated in the measurement space. This happens because the large subclass distance $d_{1,2.2,1} = ||\boldsymbol{\mu}_{1,2} - \boldsymbol{\mu}_{2,1}||^2$ dominates the MSDA criterion, and, thus, the derived projection transformation preserves well the separation of the subclasses $\mathcal{N}_{1,2}$ and $\mathcal{N}_{2,1}$, while, on the other hand, merges the two subclasses that are close together in the measurement space, $\mathcal{N}_{1,1}$ and $\mathcal{N}_{2,1}$.

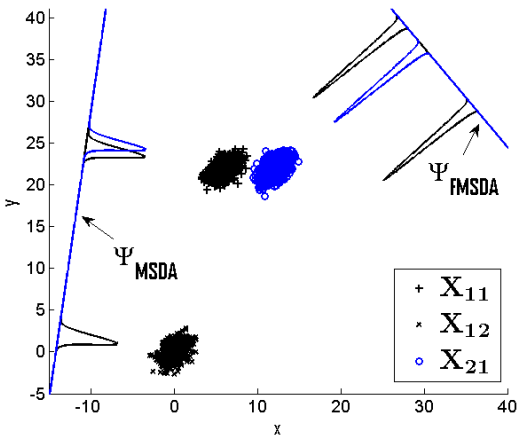

Fig. 1. *Subclass separation problem.*

To overcome the subclass separation problem, inspired from [31], we introduce the fractional-step MSDA (FMSDA) that utilizes the following objective function

$$J_{FMSDA}(\mathbf{\Psi}) = \frac{\text{tr}(\mathbf{\Psi}^T \check{\mathbf{S}}_{bsb} \mathbf{\Psi})}{\text{tr}(\mathbf{\Psi}^T \check{\mathbf{\Sigma}}_{\mathbf{X}} \mathbf{\Psi})}\ , \tag{42}$$

where the inter-between-subclass scatter matrix is modified using an appropriate weighting function $w_{i,j.k,l}$

$$\check{\mathbf{S}}_{bsb} = \sum_{i=1}^{C-1} \sum_{j=1}^{H_i} \sum_{k=i+1}^{C} \sum_{l=1}^{H_k} w_{i,j.k,l} (\boldsymbol{\mu}_{i,j} - \boldsymbol{\mu}_{k,l})(\boldsymbol{\mu}_{i,j} - \boldsymbol{\mu}_{k,l})^T\ , \tag{43}$$

and the modified covariance matrix accordingly becomes $\check{\mathbf{\Sigma}}_{\mathbf{X}} \equiv \check{\mathbf{S}}_{bsb} + \mathbf{S}_{ws}$. The weighting function is a monotonically decreasing function defined as $w_{i,j.k,l} = d_{i,j.k,l}^{-r}$, where, $d_{i,j.k,l} = ||\boldsymbol{\mu}_{i,j} - \boldsymbol{\mu}_{k,l}||^2$ is the euclidian distance between the estimated means of subclasses (i,j) and (k,l), and r is an integer number larger than two.

The FMSDA algorithm (Algorithm 2) starts with the application of the subclass partitioning procedure described in the previous section (Eqs. (37) to (41)) to derive a subclass division of the data. Then, the FMSDA criterion (42) is utilized to initialize the projection transformation matrix $\mathbf{\Psi}_D \in \mathbb{R}^{F \times D}$, and an iterative algorithm is applied, where at each iteration ρ fractional steps are used for decreasing the dimensionality of the subspace by one. That is, at the t-th fractional step of the k-th iteration the data are projected in the k-th dimensional subspace using the transformation matrix $\mathbf{\Psi}_k \in \mathbb{R}^{F \times k}$, scaled utilizing the following scaling transformation

$$\vartheta(\mathbf{y}, t) = \begin{cases} \alpha^t y_i, & i = k \\ y_i, & i = 1, \ldots, k-1, \end{cases} \tag{44}$$

where $\alpha = \exp(-\ln(\rho)/(\rho - 1))$, and the transformation matrix $\mathbf{\Psi}_k$ is recomputed using the projected and scaled data. At the end of this fractional procedure the last, k-th eigenvector of $\mathbf{\Psi}_k$ (i.e., the one that corresponds to the smallest eigenvalue of $\mathbf{\Psi}_k$) is discarded. The scaling transformation compresses the data along the direction of the last eigenvector of $\mathbf{\Psi}_k$. This allows the subclass means that are along the direction of the k-th eigenvector to be increasingly weighted in the next fractional step, causing the k-dimensional subspace to reorient so that a useful projection direction is not discarded at the end of each iteration. A validation set is used to assess the performance of the derived projection matrix $\mathbf{\Psi}_k$ at each iteration, and the one that provided the best correct classification rate (CCR) is selected.

The main advantage of FMSDA (and also EM-MSDA) over kernel variants of LDA is that the projection matrix still constitutes a linear transformation, which can provide real time performance during the testing stage. On the other hand, in contrast to EM-MSDA that tends to optimize the fit of the subclasses and simultaneously seek the projection that maximizes the inter-subclass scatter of means, FMSDA derives an initial subclass structure of the data and gradually attempts to identify the subspace that provides the best empirical recognition rate.

IV. Kernel Mixture Subclass Discriminant Analysis

The methods described in the previous sections will still not perform well when it is not possible to identify a sub-

Algorithm 2 FMSDA

Input: Annotated set $\mathbf{X}$, validation set $\mathbf{G}$, parameters ρ, r
Output: $\mathbf{\Psi}$

Initialize: $H_1 = \cdots = H_C = 1$, $H = C$, Φ_i (39), Φ (41)
repeat
  Compute class label k of class to repartition (40)
  Set $H_k \leftarrow H_k + 1$ and repartition k-th class
  Compute nongaussianity values Φ_i (39) and Φ (41)
until convergence of Φ
Compute $\mathbf{\Psi}_D$ (42), set $D = \mathrm{rank}(\check{\mathbf{S}}_{bsb})$ (43)
Set $\mathrm{CCR}_k = 0, k = 1, \ldots, D$
for $k = D$ **to** 1 **do**
  for $t = 0$ **to** $\rho - 1$ **do**
    Project training data: $\mathbf{y} = \mathbf{\Psi}_k^T \mathbf{x}$
    Apply scaling transformation: $\tilde{\mathbf{y}} = \boldsymbol{\vartheta}(\mathbf{y}, t)$
    Compute $\tilde{\mathbf{\Psi}}$ (42) using scaled data
    Set: $\mathbf{\Psi}_k \leftarrow \mathbf{\Psi}_k \tilde{\mathbf{\Psi}}$
  end for
  Discard the last (k-th) column of $\mathbf{\Psi}_k$
  Project and classify validation samples using $\mathbf{\Psi}_k$
  if sample $\mathbf{g}_j$ is classified correctly **then**
    $\mathrm{CCR}_k + +$
  end if
end for
Set: $k_o = \mathrm{argmax}_k(\mathrm{CCR}_k)$; $\mathbf{\Psi} = \mathbf{\Psi}_{k_o}$

class division that results in linearly separable classes [32], [33]. To deal with such cases, a nonlinear feature mapping $\boldsymbol{\phi}(\cdot) : \mathbb{R}^F \mapsto \mathcal{F}$ can be used to map the partitioned data into some high- or even infinite-dimensional feature space $\mathcal{F}$, where the data are expected to be linearly separable. Given a subclass partition of the data $\mathbf{X} = [\mathbf{X}_{1,1}, \ldots, \mathbf{X}_{C,H_C}]$, where $\mathbf{X}_{i,j} = [\mathbf{x}_{i,j}^1, \ldots, \mathbf{x}_{i,j}^{N_{i,j}}]$ contains the observations of the (i,j) subclass, the transformation matrix $\mathbf{W}$ that maximizes the MSDA criterion in $\mathcal{F}$ can be computed from the following generalized eigenvalue problem

$$\mathbf{S}_{bsb}^{\phi} \mathbf{W} = \check{\mathbf{\Sigma}}_{\mathbf{X}}^{\phi} \mathbf{W} \mathbf{\Lambda}^{\phi} \tag{45}$$

where,

$$\mathbf{S}_{bsb}^{\phi} = \sum_{i=1}^{C-1} \sum_{j=1}^{H_i} \sum_{k=i+1}^{C} \sum_{l=1}^{H_k} \hat{p}_{i,j} \hat{p}_{k,l} (\hat{\boldsymbol{\mu}}_{i,j}^{\phi} - \hat{\boldsymbol{\mu}}_{k,l}^{\phi})(\hat{\boldsymbol{\mu}}_{i,j}^{\phi} - \hat{\boldsymbol{\mu}}_{k,l}^{\phi})^T,$$

$$\check{\mathbf{\Sigma}}_{\mathbf{X}}^{\phi} = \mathbf{S}_{bsb}^{\phi} + \mathbf{S}_{ws}^{\phi}, \qquad \mathbf{S}_{ws}^{\phi} = \sum_{i=1}^{C} \sum_{j=1}^{H_i} \hat{p}_{i,j} \hat{\mathbf{\Sigma}}_{i,j}^{\phi},$$

are the inter-between-subclass scatter matrix, the within-subclass scatter matrix, the modified total sample covariance matrix, and $\hat{\mathbf{\Sigma}}_{i,j}^{\phi} = (1/N_{i,j}) \sum_{n=1}^{N_{i,j}} (\boldsymbol{\phi}(\mathbf{x}_{i,j}^n) - \hat{\boldsymbol{\mu}}_{i,j}^{\phi})(\boldsymbol{\phi}(\mathbf{x}_{i,j}^n) - \hat{\boldsymbol{\mu}}_{i,j}^{\phi})^T$, $\hat{\boldsymbol{\mu}}_{i,j}^{\phi} = (1/N_{i,j}) \sum_{n=1}^{N_{i,j}} \boldsymbol{\phi}(\mathbf{x}_{i,j}^n)$ are the sample covariance matrix and the sample mean of (i,j) subclass in $\mathcal{F}$ respectively. To avoid working with the mapped data explicitly (which may be impossible in case of infinite dimensional feature space $\mathcal{F}$) a kernel function formulated as an inner product in the feature space satisfying the Mercer's condition is used [9]

$$k(\mathbf{x}_{i,j}^n, \mathbf{x}_{k,l}^{\nu}) = \boldsymbol{\phi}(\mathbf{x}_{i,j}^n)^T \boldsymbol{\phi}(\mathbf{x}_{k,l}^{\nu}). \tag{46}$$

Under mild conditions, any solution of $\mathbf{W}$ must lie in the span of all the training samples [9], and, thus, it can be represented by a linear combination of the training samples as

$$\mathbf{W} = \mathbf{\Phi}(\mathbf{X}) \mathbf{\Gamma} \tag{47}$$

where $\mathbf{\Phi}(\mathbf{X}) = [\boldsymbol{\phi}(\mathbf{x}_{1,1}^1), \ldots, \boldsymbol{\phi}(\mathbf{x}_{C,H_C}^{N_{C,H_C}})]$ and $\mathbf{\Gamma} \in \mathbb{R}^{N \times C-1}$ contains the expansion coefficients. Substituting (47) into (45) and multiplying from the left with $\mathbf{\Phi}(\mathbf{X})^T$ we get $\mathbf{\Phi}^T(\mathbf{X}) \mathbf{S}_{bsb}^{\phi} \mathbf{\Phi}(\mathbf{X}) \mathbf{\Gamma} = \mathbf{\Phi}^T(\mathbf{X}) \check{\mathbf{\Sigma}}_{\mathbf{X}}^{\phi} \mathbf{\Phi}(\mathbf{X}) \mathbf{\Gamma} \mathbf{\Lambda}^{\phi}$ or

$$\mathbf{S}_{bsb}^{k} \mathbf{\Gamma} = \check{\mathbf{\Sigma}}_{\mathbf{X}}^{k} \mathbf{\Gamma} \mathbf{\Lambda}^{\phi} \tag{48}$$

where we set $\mathbf{S}_{bsb}^{k} = \mathbf{\Phi}^T(\mathbf{X}) \mathbf{S}_{bsb}^{\phi} \mathbf{\Phi}(\mathbf{X})$, $\mathbf{S}_{ws}^{k} = \mathbf{\Phi}^T(\mathbf{X}) \mathbf{S}_{ws}^{\phi} \mathbf{\Phi}(\mathbf{X})$, and $\check{\mathbf{\Sigma}}_{\mathbf{X}}^{k} = \mathbf{S}_{bsb}^{k} + \mathbf{S}_{ws}^{k}$. The mean and sample covariance matrix of the (i,j) subclass in $\mathcal{F}$ can be written in matrix product form as $\boldsymbol{\mu}_{i,j}^{\phi} = \mathbf{\Phi}(\mathbf{X}_{i,j}) \mathbf{p}_{i,j}$ and $\mathbf{\Sigma}_{i,j}^{\phi} = (1/N_{i,j}) \mathbf{\Phi}(\mathbf{X}_{i,j})(\mathbf{I} - \mathbf{P}_{i,j}) \mathbf{\Phi}^T(\mathbf{X}_{i,j})$ respectively, where, $\mathbf{p}_{i,j}$ is a $N_{i,j} \times 1$ vector and $\mathbf{P}_{i,j}$ is a $N_{i,j} \times N_{i,j}$ matrix with all elements equal to $1/N_{i,j}$. Using the above expressions, the scatter matrices in (48) can be entirely expressed by the kernel functions as follows

$$\begin{aligned} \mathbf{S}_{bsb}^{k} &= \sum_{i=1}^{C-1} \sum_{j=1}^{H_i} \sum_{k=i+1}^{C} \sum_{l=1}^{H_k} p_{i,j} p_{k,l} (\mathbf{K}_{i,j} \mathbf{p}_{i,j} - \mathbf{K}_{k,l} \mathbf{p}_{k,l}) \\ &\quad \times (\mathbf{K}_{i,j} \mathbf{p}_{i,j} - \mathbf{K}_{k,l} \mathbf{p}_{k,l})^T, \end{aligned} \tag{49}$$

$$\mathbf{S}_{ws}^{k} = \frac{1}{N} \sum_{i=1}^{C} \sum_{j=1}^{H_i} \mathbf{K}_{i,j} (\mathbf{I} - \mathbf{P}_{i,j}) \mathbf{K}_{i,j}^T \tag{50}$$

where, $\mathbf{K}_{i,j} = \mathbf{\Phi}^T(\mathbf{X}) \mathbf{\Phi}(\mathbf{X}_{i,j})$, $\mathbf{K}_{i,j} \in \mathbb{R}^{N \times N_{i,j}}$, and, thus, $\mathbf{\Gamma}$ can be easily computed from (48) using only kernel evaluations. The derived $\mathbf{\Gamma}$ can then be used for the projection of a test sample $\boldsymbol{\phi}(\mathbf{x})$ in the discriminant subspace using

$$\mathbf{z} = \mathbf{W}^T \boldsymbol{\phi}(\mathbf{x}) = \mathbf{\Gamma}^T \mathbf{k} \tag{51}$$

where $\mathbf{k} = [k(\mathbf{x}_{1,1}^1, \mathbf{x}), \ldots, k(\mathbf{x}_{C,H_C}^{N_{C,H_C}}, \mathbf{x})]^T$ and $\mathbf{z}$ is the projection of $\boldsymbol{\phi}(\mathbf{x})$.

The optimal subclass partition of the data is identified by exploiting the nongaussianity-based iterative algorithm described in Algorithms 1 and 2. Consequently, the KMSDA algorithm is presented in Algorithm 3. In certain cases, KMSDA may provide superior performance in comparison to EM-MSDA and FMSDA, however, at the cost of much higher computation time during both the training and testing stage, especially when large-scale training data sets are used (due to the large number of kernel evaluations for mapping the observations in the kernel space, and the associated computational burden of performing eigenanalysis in this space).

V. Experiments

In this section, we use 12 standard benchmarks (defining in total 19 classification tasks) to compare the proposed algorithms, EM-MSDA, FMSDA and KMSDA, with various linear and nonlinear methods, in particular with PCA [41], LDA [6], FDA [30], MDA [16], SMDA [21], SDA [17], MSDA [26], KDA [9] and KSDA [33].

Algorithm 3 KMSDA

Input: Annotated data set $\mathbf{X}$
Output: $\mathbf{\Gamma}$

1: Initialize: $H_1 = \cdots = H_C = 1$, $H = C$, Φ_i (39), Φ (41)
2: **repeat**
3: Compute class label k of class to repartition (40)
4: Set: $H_k \leftarrow H_k + 1$
5: Repartition k-th class to H_k subclasses using k-means
6: Compute Φ_i (39) and total nongaussianity Φ (41)
7: **until** convergence of Φ
8: Compute $\mathbf{\Gamma}$ (48)

A. Datasets

For the evaluation we use four datasets that belong to the UCI repository [42], two datasets from the Gunnar Rätsch's Benchmark Datasets [43], and six datasets that have been widely used for face, object and video shot detection:

Dataset 1: The Monk problem [42] is based on an artificial dataset of 432 data points in $\mathbb{N}_+^6$. Three binary classification tasks have been defined, i.e., MONK1, MONK2 and MONK3. For each task, a portion of the data has been randomly selected for forming the training set, and all 432 samples are used as the test set. In addition, in the third task 5% of the training data have been annotated wrongly in order to simulate the effect of random noise contaminating the data.

Dataset 2: The Landsat data set (LSD) consists of 6 classes (red soil, cotton crop, grey soil, damp grey soil, soil with vegetation stubble, and very damp grey soil) and 6435 feature vectors in $\mathbb{N}_+^{36}$. A partition of the dataset to training set (4435 samples) and test set (2000 samples) is already provided in [42].

Dataset 3: The Wisconsin diagnostic breast cancer (WDBC) dataset [42] is used for the recognition of benign and malignant cells from diagnostic images. This database comprises 569 diagnostic images represented in $\mathbb{R}^{30}$.

Dataset 4: The multi-feature digit dataset (MDD) [42] consists of ten classes and 200 patterns per class, i.e. 2,000 patterns in total, where each class represents one handwritten numeral ("0"-"9"). Each pattern is represented in terms of 6 feature sets, extracted from a 30×48 binary image, as follows: a) MDD-pix: 240 pixel averages in 2×3 windows, b) MDD-fou: 76 Fourier coefficients of the character shapes, c) MDD-fac: 216 profile correlations, d) MDD-kar: 64 Karhunen-Loève coefficients, e) MDD-zer: 47 Zernike moments, f) MDD-mor: 6 morphological features. Each set of features defines a separate classification task.

Dataset 5: The ETH80 database [44] consists of 8 object classes, namely, apples, pears, cars, cows, horses, dogs, tomatoes, and cups. Each class contains color images of 10 different objects recorded from 41 different views spaced evenly over the upper viewing hemisphere, i.e., the database contains 3280 images in total. In our computations the classic COIL segmentation masks of 128×128 pixels size provided in [44] are employed, resized to 25×30 pixels size and scanned column-wise to form 750-dimensional feature vectors.

Dataset 6: A subset of the MediaMill Challenge dataset is used for event recognition experiments. It consists of 492 shots belonging to one of five different sport events (baseball, basketball, football, golf, soccer). Each shot is represented by a 101-dimensional vector, where the κ-th component of this vector is in the range $[0; 1]$, expressing the degree of confidence that the κ-th concept (out of 101 concepts) is present in the shot [45]. These values are the output of SVM-based automatic concept detectors, thus represent highly-noisy data.

Datasets 7-10: Four face datasets were used in our experiments. The Sheffield face database [46] offers 575 gray-scale cropped facial images of 20 individuals, shown in a range of poses from profile to frontal views. The AT&T Database of Faces [47] contains 400 facial images of 40 individuals captured at different times, with varying lighting conditions, facial expressions, etc. The Extended Yale B (ExtYaleB) database [48] offers 2432 gray-scale cropped facial images of 38 individuals under 64 illumination conditions. The CMU Pose, Illumination, and Expression (PIE) database [49] is a collection of more than 40,000 facial images of 68 people captured across 13 different poses, under 43 different illumination conditions, and with four different expressions. For the Sheffield database, we downscaled the facial images to size 32×32 pixels resolution using bicubic interpolation, and scanned them columnwise to retrieve a set of 575 feature vectors in $\mathbb{R}^{1024}$. For the rest of the face databases we used the preprocessed 32×32 pixels resolution facial image sets of the Four Face database collection [50], [51].

Dataset 11: The Banana set [52] is a binary class dataset consisting of 5300 samples in $\mathbb{R}^2$. It is an artificial dataset created using a mixture of overlapping Gaussians.

Dataset 12: The Breast Cancer dataset [52] is a two-class dataset containing samples of 277 patients in $\mathbb{R}^9$ (excluding the nine samples that contain unknown attribute values).

B. Evaluation

A division of the datasets described in the previous subsection to training and test sets is necessary in order to evaluate the proposed algorithms. Such a division is provided along with the data for Monk and LSD. For Banana and Breast Cancer, we used 50 random realizations for training/test sets for each dataset from the Gunnar Rätsch's benchmark collection [52]. Similarly, for AT&T, ExtYaleB and PIE, 30 random realizations from the Four Face database collection [50], [51] were used, where the training set at each realization contains 10 images per subject for ExtYaleB and PIE, and 8 images per subject for AT&T. For each of the remaining datasets, we divided them following standard practices in similar works of the literature, e.g. [17], [33]. In particular, we have designed ς cross-validation (CV) folds by selecting randomly $\varpi\%$ of the samples of each class at each fold to form the test set, and used the rest of the samples as the training set. The number of folds ς and the percentage of test samples $\varpi\%$ for WDBC, MDD, ETH-80, Sheffield, and Mediamill dataset were set to $(\varsigma, \varpi) = (1, 50), (5, 50), (10, 10), (30, 60)$ and $(30, 20)$ respectively.

The optimal parameters of each method at each CV fold are selected using as primary metric the correct classification rate

(CCR). For this, the global-to-local search strategy is applied (e.g., see [9]), i.e., after globally searching using a coarse scale of the parameter space, a candidate interval where the optimal parameters might exist is retrieved, and then a finer inspection for identifying the optimal parameters within this interval is performed. For the subclass methods (SDA, MSDA, FMSDA, EM-MSDA, KMSDA) we optimize over the number of subclasses in each class, and consequently over the total number of subclasses. For the FMSDA method we additionally require the identification of the exponent r of the weighting functions in (43) and the number of fractional steps $\rho \in \mathbb{N}_+$ for decreasing the subspace dimensionality by one. For the optimization of these parameters we search over the following values: $r = 3, 4, \ldots, 16$ and $\rho = 3, 4, \ldots, 20$. Similarly, for the kernel-based methods (KDA, KSDA, KMSDA) we need to identify the optimal parameters of the kernel functions. In our experiments we used two types of base kernels: Guassian radial basis function $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|/2\sigma^2)$, $\sigma \in \mathbb{R}_+$, and the polynomial function $k(\mathbf{x}_i, \mathbf{x}_j) = ((\mathbf{x}_i \mathbf{x}_j) + o)^\varrho$, $o \in \mathbb{R}$, $\varrho \in \mathbb{N}_+$. For their parameters we search for the optimal values over the following ranges: $o = 0, 1$, $\varrho = 1, 2, \ldots, 8$, $\sigma = 0.1, 0.2, \ldots, 4$. We should also note that for the datasets whose number of training observations N is small compared to their dimensionality F (such as the Sheffield and ETH-80 datasets), the computation of the inverse of the MLE of the sample covariance matrix (16) by the EM-based methods, for instance SMDA and EM-MSDA, will be especially problematic (e.g. see [2], [53]). In these cases, we compute the inverse using the eigenvalue decomposition of the sample covariance matrix, keeping only the eigenvalue components whose eigenvalues are above a specific threshold [2].

The recognition performance of a method regarding a dataset is measured using the average CCR (ACCR) along all CV folds, i.e., at each CV fold the maximum correct classification rate (CCR) for the different set of parameters is retained, and the CCRs are averaged along all CV folds. Similarly, the ground truth labels and the predicted labels at each CV fold for each algorithm are retained, and the McNemar's hypothesis test [54], [55] with a significance level of 0.025 is used to evaluate the statistical significance of the difference in the performance between each of the proposed algorithms and any other algorithm used in our experiments. Moreover, to compare the computational complexity of the algorithms we recorded the testing and training times in minutes, on a Intel i7 2.8GHz machine, with respect to one CV fold for each method and each dataset. Except for MDA and SMDA, for which their R package implementation [21] is exploited, all the other algorithms are compared using an unoptimized Matlab implementation. The FMSDA algorithm was then used as the baseline algorithm to compute the speedup rate s_κ for the κ-th algorithm using $s_\kappa = T_{fmsda}/T_\kappa$, where T_{fmsda} and T_κ are the training (or testing) time concerning the FMSDA and the κ-th algorithm respectively.

The ACCRs of the methods along with the average dimensionality in the discriminant subspace are shown in Table I, while, the results of the statistical significance tests are shown in Table II. In the latter, a cell contains the symbol $+, \star$ or $\sim$ for FMSDA, EMMSDA or KMSDA respectively, in order to denote that the improvement in performance achieved by the aforementioned methods in comparison to the method corresponding to the column of the table is statistically significant. Finally, the speedup rate for the training stage (left side of the comma) and testing stage (right side of the comma) of the algorithms on each dataset are depicted in Table III, where higher speedup values indicate faster computations. In every table we have divided the methods into three groups, namely, linear, subclass and kernel-based methods. With respect to this partitioning, for Tables I and III we have used bold digits and underlined-bold digits to denote the best performance rate within each group and along all methods respectively.

From Table I we can see that for the majority of the datasets the best ACCR among the linear subclass methods is provided by FMSDA (in 10 out of 19 classification tasks of Table I) or EM-MSDA (again in 10 out of 19 tasks). In overall the best ACCR among all methods is achieved by KMSDA (in 17 out of 19 tasks). We should also note that in many cases FMSDA and EM-MSDA outperform the kernel-based methods as well (including KMSDA in 2 classification tasks, while they match KMSDA's performance in another 2 tasks). Between FMSDA and EM-MSDA, we observe that the former tends to perform better when the data dimensionality is larger than the number of the samples, and at the same time many subclasses are necessary in order to capture the subclass structure of the data. In these cases, the training samples per subclass are limited and consequently the subclass covariance matrices are poorly estimated [53]. This adversely affects the performance of EM-based methods. For instance, the performance of SMDA and EM-MSDA on the ETH80 dataset (which contains 8 object classes and each object class 10 different objects) is considerably lower than that of FMSDA.

From the results in Table II we can also see that the performance improvements attained by the proposed methods are statistically significant for most of the datasets. From Table III, we additionally see that FMSDA and EM-MSDA provide systematically lower computation times during the testing stage among all linear subclass methods (and, as expected are also faster than the kernel-based methods). This quality of FMSDA and EM-MSDA is a critical advantage of them, especially for applications that require real-time or near real-time processing of large data volumes, such as event detection in video streams. Summarizing, we observe that the three proposed methods in most cases outperform the current state of the art as recently reported for KSDA in [33] and in the also very recent works [21], [26], at the same time offering competitive response times during the testing (recognition) stage.

VI. Conclusions

Subclass DA methods are attractive alternatives to the kernel DA variants because they offer fast (often real-time) computations and comparable recognition performance. Furthermore, combining subclass partitioning and the kernel trick in a single DA method opens new possibilities for improved DA effectiveness. MSDA is a very recent subclass method, that utilizes an effective partitioning procedure to derive a

TABLE I
RECOGNITION RATES (ACCRS). THE DIMENSIONALITY OF EACH DATASET IS GIVEN IN THE PARENTHESIS NEXT TO THE DATASET'S NAME.

	Linear methods			*Linear subclass methods*						*Kernel methods*		
	PCA	*LDA*	*FDA*	*MDA*	*SMDA*	*SDA*	*MSDA*	*FMSDA*	*EM-MSDA*	*KDA*	*KSDA*	*KMSDA*
MONK1 (6)	**81.4%** (6)	69.9% (1)	69.9% (1)	95.6% (6)	93% (6)	84.4% (6)	98.8% (4)	**99.8%** (3)	96.2% (3)	90.2% (1)	94.4% (5)	**99.8%** (3)
MONK2 (6)	**71.7%** (6)	67.3% (1)	67.3% (1)	76.1% (5)	65.3% (3)	84.9% (6)	87.2% (6)	85.2% (5)	**90%** (6)	81.9% (1)	83.5% (29)	**91%** (16)
MONK3 (6)	**86.8%** (6)	85.8% (1)	85.8% (1)	94.1% (6)	89.5% (6)	88.9% (3)	93.0% (3)	**94.4%** (2)	**94.4%** (3)	95.8% (1)	94.6% (3)	**96.2%** (1)
LSD (36)	**89.4%** (36)	84% (5)	85.2% (5)	88.1% (35)	83.2% (35)	87.8% (17)	89.9% (16)	**90.4%** (17)	89% (19)	87.1% (5)	89.3% (6)	**89.9%** (38)
WDBC (30)	89.4% (30)	93.3% (1)	**94.7%** (1)	96.1% (27)	91.9% (19)	94.7% (15)	96.8% (10)	97.1% (5)	**97.2%** (8)	93.3% (1)	94.3% (5)	**95.4%** (5)
MDD-pix (240)	**97.5%** (240)	94.1% (9)	95.3% (8)	96.4% (105)	96.7% (71)	96.6% (55)	**97.8%** (70)	97.3% (44)	95.9% (19)	98.5% (9)	98.2% (47)	**98.6%** (23)
MDD-fou (76)	**82.6%** (76)	80.4% (9)	80.2% (8)	82.9% (72)	82.6% (64)	81.3% (52)	82.2% (24)	82.7% (13)	**83.1%** (23)	85.4% (9)	84% (17)	**85.9%** (12)
MDD-fac (216)	94.4% (213)	**97.6%** (9)	**97.6%** (8)	98.2% (63)	97.2% (80)	96.2% (31)	98.3% (29)	**98.4%** (20)	**98.4%** (43)	98.2% (9)	95.7% (23)	**98.8%** (13)
MDD-kar (64)	**97.2%** (64)	95.7% (9)	95.6% (8)	97.2% (64)	97.3% (55)	97.0% (41)	97.2% (20)	96.8% (15)	**97.5%** (57)	98.3% (9)	98.1% (23)	**98.6%** (22)
MDD-zer (47)	**81.1%** (47)	76.4% (9)	79.3% (8)	83% (47)	**83.2%** (24)	81.2% (22)	**83.2%** (22)	82.8% (27)	78.7% (16)	84.0% (9)	83% (23)	**84.8%** (27)
MDD-mor (6)	59.5% (6)	67.2% (6)	**67.3%** (4)	67.4% (6)	68.2% (6)	67.5% (6)	68.3% (6)	**68.5%** (5)	67.6% (6)	65.8% (9)	64.8% (17)	**68.7%** (33)
ETH80 (750)	**83.5%** (750)	69.5% (7)	77.1% (7)	79.4% (115)	73.6% (82)	85.1% (73)	86.5% (75)	**87%** (65)	74.7% (41)	86.9% (7)	83.2% (17)	**87.2%** (63)
Mediamill (101)	**68%** (101)	64.9% (4)	63.9% (4)	71.3% (21)	60.2% (9)	69.3% (35)	71.4 (23)	**75.8%** (31)	72% (27)	74.3% (4)	76.8% (26)	**77.5%** (24)
Sheffield (1024)	94.9% (236)	95.5% (19)	**96.8%** (19)	90.3% (39)	95.2% (74)	97.2% (31)	97.6% (24)	**98.5%** (18)	97.3% (34)	98% (19)	**98.5%** (31)	**98.5%** (23)
A&T (1024)	92.5% (319)	96.1% (39)	**97.8%** (33)	93.5% (81)	95.4% (81)	97.7% (47)	98.3% (44)	**98.9%** (29)	**98.9%** (39)	97.9% (39)	98.9% (47)	**99.4%** (40)
ExtYaleB (1024)	53.4% (379)	**85.6%** (37)	**85.6%** (35)	73.3% (75)	66.2% (39)	86.5% (71)	86.7% (48)	87.7% (40)	**87.8%** (37)	85.5% (37)	87.9% (48)	**89.9%** (39)
PIE (1024)	44.4% (679)	77.6% (67)	**79.8%** (41)	82.1% (105)	62.6% (69)	80% (67)	84.9% (85)	**86.5%** (66)	86.1% (67)	85.1% (62)	83.7% (67)	**86.7%** (74)
Banana (1024)	**68.4%** (2)	57.2% (1)	63.7% (1)	88.3% (4)	88.3% (2)	86.2% (2)	88.4% (2)	88.3% (2)	**88.9%** (2)	88.8% (1)	88.4% (3)	**89.5%** (4)
B. Cancer (1024)	**67.6%** (9)	65.2% (1)	65.2% (1)	71.4% (4)	69.4% (2)	70.4% (4)	73.9% (7)	75.3% (7)	**77.9%** (9)	74.7% (1)	74.6% (3)	**79.2%** (3)

TABLE II
STATISTICAL SIGNIFICANCE RESULTS. SYMBOLS $+$, $*$, $\sim$ DENOTE THAT THE IMPROVEMENT IN PERFORMANCE ACHIEVED BY FMSDA ($+$), EMMSDA ($*$), OR KMSDA ($\sim$) IN COMPARISON TO THE METHOD OF A GIVEN COLUMN IS STATISTICALLY SIGNIFICANT.

	Linear methods			*Linear subclass methods*				*Kernel methods*	
	PCA	*LDA*	*FDA*	*MDA*	*SMDA*	*SDA*	*MSDA*	*KDA*	*KSDA*
MONK1	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$+, \sim$	$+, \sim$	$+, *, \sim$		$+, *, \sim$	$+, \sim$
MONK2	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$*, \sim$		$*, \sim$	$*, \sim$
MONK3	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$		$+, *, \sim$	$+, *, \sim$			
LSD		$+, *, \sim$	$+, *, \sim$		$+, *, \sim$			$+, \sim$	
WDBC	$+, *, \sim$				$+, *$				
MDD-pix	$\sim$	$+, \sim$	$\sim$	$\sim$	$\sim$	$\sim$			
MDD-fou	$\sim$	$+, *, \sim$	$+, *, \sim$	$\sim$	$\sim$	$+, *, \sim$	$\sim$		$\sim$
MDD-fac	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$\sim$	$+, *, \sim$	$+, *, \sim$	$\sim$	$\sim$	$+, *, \sim$
MDD-kar	$\sim$	$+, *, \sim$	$+, *, \sim$	$\sim$	$\sim$	$*, \sim$	$\sim$	$\sim$	
MDD-zer	$+, \sim$	$+, *, \sim$	$+, \sim$	$\sim$	$\sim$	$+, \sim$	$\sim$		$\sim$
MDD-mor	$+, *, \sim$	$+, \sim$	$+, \sim$	$+, \sim$		$+, \sim$		$+, \sim$	$+, \sim$
ETH80	$+, \sim$	$+, *, \sim$	$+, \sim$	$+, \sim$	$+, \sim$	$+, \sim$			$+, \sim$
Mediamill	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$+, *$	$+, *, \sim$	$+, *$	$+, *$	$+, *$	
Sheffield	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$+, \sim$	$+, \sim$	$+, \sim$	
A&T	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$\sim$	$+, *, \sim$	$\sim$
ExtYaleB	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$\sim$
PIE	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$
Banana	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$*, \sim$	$*, \sim$	$+, *, \sim$
B. Cancer	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	$+, *, \sim$	

TABLE III
TRAINING AND TESTING SPEEDUP RATES.

	Linear methods			*Linear subclass methods*						*Kernel methods*		
	PCA	*LDA*	*FDA*	*MDA*	*SMDA*	*SDA*	*MSDA*	*FMSDA*	*EM-MSDA*	*KDA*	*KSDA*	*KMSDA*
MONK1	15.19, 0.13	**34.07**, **1**	1.79, 0.89	**11.81**, 0.56	4.20, 0.57	6.74, 0.98	3.51, 1.09	1, **1**	2.87, 0.92	**3.08**, 0.59	0.20, **0.65**	0.18, 0.64
MONK2	10.19, 0.12	**35.74**, **0.94**	2.07, 0.88	**12.12**, 0.58	7.84, 0.45	8.98, 0.94	4.46, 0.94	1, 1	1.32, **1.18**	**3.18**, **0.59**	0.13, 0.55	0.13, 0.58
MONK3	15.96, 0.13	**35.86**, **1**	2.13, 0.91	**12.7**, 0.83	4.45, 0.65	8.74, 0.99	4.05, 0.95	1, 1	3.54, **1.25**	**3.03**, 0.54	0.21, **0.65**	0.19, 0.62
LSD	331.16, 0.76	**564.22**, 1.32	8.68, **1.42**	**37.59**, 0.98	6.71, 0.64	30.89, 0.99	9.30, 0.97	1, **1**	2.20, 0.96	**0.07**, **0.52**	0.01, 0.46	0.01, 0.43
WDBC	23.10, 0.10	**54.40**, **1.06**	2.60, 0.76	**5.75**, 0.17	0.18, 0.15	4.85, 0.94	3.43, 0.99	1, 1	1.32, **1.06**	**1.29**, 0.45	0.03, 0.46	0.03, **0.47**
MDD-pix	410.92, 0.33	**1136.07**, **1.63**	6.04, 1.41	13.41, 0.06	1.1, 0.37	**14.86**, 0.73	1.29, 0.78	1, 1	3.57, **1.13**	**3.02**, **0.39**	0.17, 0.35	0.03, 0.35
MDD-fou	517.16, 0.65	**934.62**, **1.27**	3.23, 1.20	9.18, 0.07	1.18, 0.61	**20.41**, 0.61	6.46, 0.99	1, **1**	3.68, 0.98	**1.31**, **0.38**	0.07, 0.35	0.01, 0.34
MDD-fac	169.54, 0.30	**357.92**, **1.40**	2.22, 1.24	4.06, 0.05	0.39, 0.4	**6.44**, 0.78	1.61, 0.95	1, **1**	1.35, 0.96	**1.05**, **0.34**	0.06, 0.31	0.01, 0.31
MDD-kar	181.93, 0.65	**339.60**, **1.27**	1.06, 1.11	2.96, 0.07	0.48, 0.62	**7.96**, 0.60	2.68, 0.91	1, **1**	1.07, 0.85	**0.39**, **0.36**	0.02, 0.32	0.01, 0.32
MDD-zer	212.70, 0.80	**296.79**, 1.09	1.22, **1.13**	3.62, 0.09	0.79, 0.89	**10.64**, 0.93	2.13, 0.88	1, **1**	3.40, 0.96	**0.46**, **0.4**	0.02, 0.34	0.01, 0.34
MDD-mor	**389.80**, 1.03	347.17, 1.06	1.85, **1.13**	6.41, 0.15	8.13, 0.21	**15.87**, 0.87	0.65, 0.37	1, 1	3.09, 0.82	**0.34**, 0.33	0.02, **0.35**	0.01, 0.30
ETH80	380.92, 0.09	**1123.13**, 1.19	13.31, **2.09**	**21.85**, 0.23	1.12, 1.07	7.14, 1.02	1.18, 1.05	1, 1	0.51, **1.08**	**1.37**, **0.32**	0.03, 0.28	0.01, 0.28
Mediamill	168.18, 0.06	**305.82**, **1.78**	5.47, 1.17	19.42, 0.1	64.88, 0.41	**21.80**, 0.92	10, 1.02	1, 1	2.13, **1.04**	**4.32**, **0.53**	0.14, 0.47	0.05, 0.47
Sheffield	470.65, 0.12	**678.09**, **0.99**	1.09, 0.58	**62.25**, 0.08	3.72, 0.06	6.53, 0.26	18.14, 0.98	1, **1**	2.52, 0.99	**74.04**, **0.39**	12.33, 0.38	4.30, **0.39**
A&T	2470.1, 0.31	**3169.4**, **0.93**	0.5, 0.87	**131.33**, 0.09	1.95, 0.19	1.47, 0.79	0.83, 0.82	1, **1**	0.61, 0.93	**27.34**, **0.36**	1.38, 0.25	1.29, 0.34
ExtYaleB	510.28, 0.27	**919.18**, 0.92	0.72, **1.13**	**22.76**, 0.09	3.65, 0.09	6.29, 0.66	1.32, 0.74	1, 1	0.59, **1.02**	**16.47**, **0.47**	3.17, 0.36	1.53, 0.4
PIE	472.67, 0.12	**2115.13**, 0.97	0.47, **1.03**	**54.84**, 0.38	5.77, 0.31	13.36, 0.99	0.61, 0.74	1, **1**	1.81, **1**	**44.09**, **0.49**	2.22, 0.43	1.39, 0.34
Banana	**11.98**, 0.93	7.65, **1.1**	0.74, **1.1**	**50.79**, 0.54	15.63, 0.83	2.29, 0.99	1.34, 0.99	1, **1**	0.59, 0.98	**0.17**, **0.69**	0.04, 0.59	0.03, 0.49
B. Cancer	21.67, 0.4	**40.2**, **1.17**	3.14, 1.14	**37.8**, 0.23	3.37, 0.63	21.07, **1.04**	9.06, 0.94	1, 1	0.33, 0.90	**0.43**, **0.25**	0.27, 0.18	0.08, 0.18

Gaussian homoscedastic division of the data. In this work, we extended MSDA in three different ways: a) EM-MSDA was derived by linking MSDA with the Guassian mixture model, b) FMSDA was proposed in order to solve the so-called subclass separation problem, and c) KMSDA was presented for separating categories with nonlinearly separable subclasses using the kernel trick. The effectiveness of the three proposed DA methods was verified by extensive experimentation on various publicly available standard benchmarks.

Our methods could also be extended and used in additional related problems, such as feature selection. Typically, this is an application domain for methods such as support vector machines [56]. Recently, a feature selection method based on LDA was proposed in [57]. This method ranks each feature using the sum of the eigenvectors of the LDA projection matrix. In a similar fashion FMSDA could be easily modified to rank and discard instead of the least discriminant dimension the least significant feature at each iteration. We plan to investigate this possibility, as well the possibility of extending the proposed methods for signature-based classification, taking advantage of the work described in [58].

ACKNOWLEDGMENT

This work was supported by the EC under contracts FP7-248984 GLOCAL and FP7-287911 LinkedTV.

APPENDIX A
DERIVATION OF EQUATIONS IN SECTION II

A. Derivation of Eqs. (6) and (7)

The Gaussian mixture distribution concerning the i-th class in (5) can be derived in terms of latent variables [36], [37], as described in the following. Let $Z_i \in \mathbb{R}^{H_i}$ be a categorical latent random vector concerning the i-th class, whose parameter space $\mathcal{Z}_i$ is the standard base of $\mathbb{R}^{H_i}$, i.e., $\mathcal{Z}_i = \{\mathbf{e}_{i,1}, \ldots, \mathbf{e}_{i,H_i}\}$, where only the j-th element of the unit vector $\mathbf{e}_{i,j}$ is equal to one and all other elements are equal to zero. Setting $p(Z_i = \mathbf{e}_{i,j}) = \pi_{i,j}$ and $p(\mathbf{x}|Z_i = \mathbf{e}_{i,j}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{i,j})$ the marginal and conditional densities, $p(\mathbf{z}_i)$ and $p(\mathbf{x}|\mathbf{z}_i)$, are expressed in terms of the mixing coefficients and mixture components respectively, $p(\mathbf{z}_i) = \prod_{j=1}^{H_i} \pi_{i,j}^{z_{i,j}}$, $p(\mathbf{x}|\mathbf{z}_i) = \prod_{j=1}^{H_i} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{i,j})^{z_{i,j}}$. Thus, using the product rule of probability we can express the i-th class-conditional joint density as

$$\begin{aligned} p(\mathbf{x}, \mathbf{z}_i|\omega_i) &= p(\mathbf{z}_i|\omega_i)p(\mathbf{x}|\mathbf{z}_i, \omega_i) = p(\mathbf{z}_i)p(\mathbf{x}|\mathbf{z}_i) \\ &= \prod_{j=1}^{H_i} (\pi_{i,j}\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{i,j}))^{z_{i,j}}, \end{aligned} \tag{52}$$

where we have used the fact that $\mathbf{x}$ is conditionally independent of ω_i given $\mathbf{z}_i$, and $\mathbf{z}_i$ is independent of ω_i. The i-th class-conditional marginal distribution of $\mathbf{x}$ can then be written as

$$p(\mathbf{x}|\omega_i) = \sum_{\mathbf{z}_i} p(\mathbf{x}, \mathbf{z}_i|\omega_i) = \sum_{j=1}^{H_i} \pi_{i,j}\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{i,j}), \tag{53}$$

which is a Gaussian mixture equivalent to (5), and, using the Bayes' rule the posterior distribution is also derived

$$p(\mathbf{z}_i|\mathbf{x}, \omega_i) = \frac{\prod_{j=1}^{H_i} (\pi_{i,j}\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{i,j}))^{z_{i,j}}}{\sum_{j=1}^{H_i} \pi_{i,j}\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{i,j})}. \tag{54}$$

Therefore, under the i.i.d. assumption, the likelihood of the complete data set is expressed as (p.108, [27])

$$\begin{aligned} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) &= \prod_{i=1}^{C} \prod_{n=1}^{N_i} p(\mathbf{x}_i^n, \mathbf{z}_i^n|\omega_i) \\ &= \prod_{i=1}^{C} \prod_{n=1}^{N_i} \prod_{j=1}^{H_i} (\pi_{i,j}\mathcal{N}(\mathbf{x}_i^n|\boldsymbol{\mu}_{i,j}))^{z_{i,j}^n}. \end{aligned} \tag{55}$$

while the posterior distribution takes the form

$$p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}) \propto \prod_{i=1}^{C} \prod_{n=1}^{N_i} \prod_{j=1}^{H_i} (\pi_{i,j}\mathcal{N}(\mathbf{x}_i^n|\boldsymbol{\mu}_{i,j}))^{z_{i,j}^n}, \tag{56}$$

where $\mathbf{Z} = \{\mathbf{Z}_1, \ldots, \mathbf{Z}_C\}$ is the set of all categorical vectors. Observing that the posterior distribution is independent over $z_{i,j}^n$, the expectation of the categorical variables can be derived

$$\mathbb{E}[z_{i,j}^n] = \frac{\sum_{j=1}^{H_i} z_{i,j}^n (\pi_{i,j}\mathcal{N}(\mathbf{x}_i^n|\boldsymbol{\mu}_{i,j}))^{z_{i,j}^n}}{\sum_{j=1}^{H_i} \pi_{i,j}\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{i,j})}, \tag{57}$$

and simplifying the above, we arrive to the definition of the responsibilities in (7).

Moreover, from (56) the log likelihood of the complete data set is retrieved

$$\ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) = \sum_{i=1}^{C} \sum_{n=1}^{N_i} \sum_{j=1}^{H_i} z_{i,j}^n (\ln \pi_{i,j} + \ln \mathcal{N}(\mathbf{x}_i^n|\boldsymbol{\mu}_{i,j})). \tag{58}$$

Applying the expectation operator to the above expression and substituting $\mathbb{E}[z_{i,j,n}]$ from (7) the expectation of the complete data log-likelihood is expressed as

$$\begin{aligned} &\mathbb{E}[\ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})] \\ &= \sum_{i=1}^{C} \sum_{n=1}^{N_i} \sum_{j=1}^{H_i} h_{i,j}^n (\ln \pi_{i,j} + \ln \mathcal{N}(\mathbf{x}_{i,n}|\boldsymbol{\mu}_{i,j}, \boldsymbol{\Sigma})) \\ &= \sum_{i=1}^{C} \sum_{j=1}^{H_i} \tilde{N}_{i,j} \ln \pi_{i,j} - \frac{NF}{2} \ln(2\pi) + \frac{N}{2} \ln |\boldsymbol{\Sigma}^{-1}| \\ &\quad -\frac{1}{2} \sum_{i=1}^{C} \sum_{n=1}^{N_i} \sum_{j=1}^{H_i} h_{i,j,n} (\mathbf{x}_{i,n} - \boldsymbol{\mu}_{i,j})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_{i,n} - \boldsymbol{\mu}_{i,j}). \end{aligned} \tag{59}$$

Using the identity $(\mathbf{x}_i^n - \boldsymbol{\mu}_{i,j})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_i^n - \boldsymbol{\mu}_{i,j}) = (\mathbf{x}_i^n - \bar{\mathbf{x}}_i^n)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_i^n - \bar{\mathbf{x}}_{i,j}) + (\bar{\mathbf{x}}_{i,j} - \boldsymbol{\mu}_{i,j})^T \boldsymbol{\Sigma}^{-1} (\bar{\mathbf{x}}_{i,j} - \boldsymbol{\mu}_{i,j}) + 2(\mathbf{x}_i^n - \bar{\mathbf{x}}_{i,j})^T \boldsymbol{\Sigma}^{-1} (\bar{\mathbf{x}}_{i,j} - \boldsymbol{\mu}_{i,j})$ along with the fact that $\sum_{n=1}^{N_i} (\mathbf{x}_i^n - \bar{\mathbf{x}}_{i,j})^T \boldsymbol{\Sigma}^{-1} (\bar{\mathbf{x}}_{i,j} - \boldsymbol{\mu}_{i,j}) = 0$, and multiplying both sides by two, we arrive to (6).

B. Derivation of Eq. (18)

The constraint that the mixing coefficients should sum to one can be incorporated in (17) using C lagrange multipliers $\eta_i, i = 1, \ldots, C$. Therefore, we need to find the stationary point of

$$\begin{aligned} &\sum_{i=1}^{C} \sum_{n=1}^{N_i} \sum_{j=1}^{H_i} h_{i,j}^n (\ln \pi_{i,j} + \ln \mathcal{N}(\mathbf{x}_i^n|\boldsymbol{\mu}_{i,j})) \\ &\qquad + \sum_{i=1}^{C} \eta_i (\sum_{j=1}^{H_i} \pi_{i,j} - 1) \end{aligned} \tag{60}$$

with respect to $\pi_{i,j}$ and η_i. Optimizing over $\pi_{i,j}$ we arrive to $\tilde{N}_{i,j}/\pi_{i,j} + \eta_i = 0$. If we multiply both sides with $\pi_{i,j}$ and sum over all subclasses of the i-th class we get $\eta_i = -N_i$. Eliminating η_i we obtain (18).

REFERENCES

[1] Y. Ephraim and H. L. V. Trees, "A signal subspace approach for speech enhancement," *IEEE Trans. Speech Audio Process.*, vol. 3, no. 4, pp. 251–266, Jul. 1995.

[2] K. Fukunaga, *Introduction to statistical pattern recognition (2nd ed.)*. San Diego, CA, USA: Academic Press Professional, Inc., 1990.

[3] R. Duda, P. Hart, and D. Stork, *Pattern Classification, (2nd ed.)*. New York, USA: John Wiley & Sons, Inc., 2001.

[4] S. Ji and J. Ye, "Generalized linear discriminant analysis: A unified framework and efficient model selection," *IEEE Trans. Neural Netw.*, vol. 19, no. 10, pp. 1768–1782, Oct. 2008.

[5] C. B. Moler and G. W. Stewart, "An algorithm for generalized matrix eigenvalue problems," *SIAM Journal on Numerical Analysis*, vol. 10, no. 2, pp. 241–256, Apr. 1973.

[6] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 1997.

[7] H. Li, T. Jiang, and K. Zhang, "Efficient and robust feature extraction by maximum margin criterion," *IEEE Trans. Neural Netw.*, vol. 17, no. 1, pp. 157–165, Oct. 2006.

[8] C. S. Dhir and S.-Y. Lee, "Discriminant independent component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 6, pp. 845–857, Jun. 2011.

[9] K. Muller, S. Mika, G. Ratsch, S. Tsuda, and B. Scholkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 181–202, Mar. 2001.

[10] L. Wang, K. L. Chan, P. Xue, and L. Zhou, "A kernel-induced space selection approach to model selection in KLDA," *IEEE Trans. Neural Netw.*, vol. 19, no. 12, pp. 2116–2131, Dec. 2008.

[11] Z. Li, D. Lin, and X. Tang, "Nonparametric discriminant analysis for face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 4, pp. 755–761, Apr. 2009.

[12] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Face recognition using kernel direct discriminant analysis algorithms," *IEEE Trans. Neural Netw.*, vol. 14, no. 1, pp. 117–126, Jan. 2003.

[13] 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. and Learning Syst.*, vol. 23, no. 3, pp. 526–534, Mar. 2012.

[14] Z. Fan, Y. Xu, and D. Zhang, "Local linear discriminant analysis framework using sample neighbors," *IEEE Trans. Neural Netw.*, vol. 22, no. 7, pp. 1119–1132, Jul. 2011.

[15] B.-C. Kuo and K.-Y. Chang, "Feature extractions for small sample size classification problem," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 3, pp. 756–764, Mar. 2007.

[16] T. Hastie and R. Tibshirani, "Discriminant analysis by Gaussian mixtures," *Journal of the Royal Statistical Society. Series B*, vol. 58, no. 1, pp. 155–176, Jul. 1996.

[17] M. Zhu and A. Martinez, "Subclass discriminant analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 8, pp. 1274–1286, Aug. 2006.

[18] S.-W. Kim and R. P. W. Duin, "On using a pre-clustering technique to optimize LDA-based classifiers for appearance-based face recognition," in *Proc. 12th Iberoamerican Congress on Pattern Recognition*, Vina del Mar-Valparaiso, Chile, Nov. 2007, pp. 466–476.

[19] M. H. Yang, D. J. Kriegman, and N. Ahuja, "Face detection using multimodal density models," *Computer Vision and Image Understanding*, vol. 84, no. 2, pp. 264–284, Oct. 2001.

[20] A. Pnevmatikakis and L. Polymenakos, "Subclass linear discriminant analysis for video-based face recognition," *Journal of Visual Communication and Image Representation*, vol. 20, no. 8, pp. 543–551, Nov. 2009.

[21] L. Clemmensen, T. Hastie, D. Witten, and B. Ersboll, "Sparse discriminant analysis," *Technometrics*, vol. 53, no. 4, pp. 406–413, Nov. 2011.

[22] D. Wu and K. L. Boyer, "Resilient subclass discriminant analysis," in *Proc. IEEE Int. Conf. on Computer Vision (ICCV 2009)*, Kyoto, Japan, Sep./Oct. 2009, pp. 389–396.

[23] F. Oveisi, "Subclass discriminant analysis using dynamic cluster formation for EEG-based brain-computer interface," in *Proc. IEEE/EMBS 4th Int. Conf. on Neural Engineering*, Antalya, Turkey, May 2009, pp. 303–306.

[24] S.-W. Kim, "A pre-clustering technique for optimizing subclass discriminant analysis," *Pattern Recogn. Lett.*, vol. 31, no. 6, pp. 462–468, Apr. 2010.

[25] N. Gkalelis, V. Mezaris, and I. Kompatsiaris, "High-level event detection in video exploiting discriminant concepts," in *Proc. 9th International Workshop on Content-Based Multimedia Indexing (CBMI 2011)*, Madrid, Spain, Jun. 2011, pp. 85–90.

[26] ——, "Mixture subclass discriminant analysis," *IEEE Signal Process. Lett.*, vol. 18, no. 5, pp. 319–332, May 2011.

[27] K. V. Mardia, J. T. Kent, and J. M. Bibby, *Multivariate analysis*. Academic Press, 1979.

[28] N. A. Campbell, "Canonical variate analysis - A general model formulation," *Australian & New Zealand Journal of Statistics*, vol. 26, no. 1, pp. 86–96, 1984.

[29] D. Tao, X. Li, X. Wu, and S. J. Maybank, "Geometric mean for subspace selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 260–274, Feb. 2009.

[30] R. Lotlikar and R. Kothari, "Fractional-step dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 6, pp. 623–627, Jun. 2000.

[31] M. Loog, R. P. W. Duin, and R. Haeb-Umbach, "Multiclass linear dimension reduction by weighted pairwise Fisher criteria," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 7, pp. 762–766, Jul. 2001.

[32] B. Chen, L. Yuan, H. Liu, and Z. Bao, "Kernel subclass discriminant analysis," *Neurocomputing*, vol. 71, no. 1–3, pp. 455–458, Dec. 2007.

[33] D. You, O. C. Hamsici, and A. M. Martinez, "Kernel optimization in discriminant analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 631–638, Mar. 2011.

[34] V. Vapnik, *Statistical learning theory*. New York: Willey, 1998.

[35] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[36] G. Mclachlan and D. Peel, *Finite Mixture Models*. New Work: Wiley-Interscience, 2000.

[37] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[38] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*. New York, USA: Springer, 1996.

[39] N. Vlassis and A. Likas, "A kurtosis-based dynamic approach to gaussian mixture modeling," *IEEE Trans. Syst., Man, Cybern. A*, vol. 29, no. 4, pp. 393–399, Jul. 1999.

[40] L. Wang and J. Ma, "A kurtosis and skewness based criterion for model selection on gaussian mixture," in *2nd Int. Conf. on BioMedical Engineering and Informatics*, Tianjin, China, Oct. 2009, pp. 1–5.

[41] I. T. Jolliffe, *Principal Component Analysis*. New York, USA: Springer, Oct. 2002.

[42] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml

[43] "Gunnar raetsch's benchmark datasets," http://theoval.cmp.uea.ac.uk/~gcc/matlab/default.html#benchmarks, accessed 2012-05-01.

[44] B. Leibe and B. Schiele, "Analyzing appearance and contour based methods for object categorization," in *Proc. 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, Madison, WI, USA, Jun. 2003, pp. II–409–15.

[45] N. Gkalelis, V. Mezaris, and I. Kompatsiaris, "Automatic event-based indexing of multimedia content using a joint content-event model," in *ACM Multimedia 2010, (EiMM10)*, Firenze, Italy, Oct. 2010.

[46] D. B. Graham and N. M. Allinson, "Characterizing virtual eigensignatures for general purpose face recognition," in *Face Recognition: From Theory to Applications, Computer and Systems Sciences*, H. Wechsler et al., Ed. NATO ASI Series F, 1998, vol. 163, pp. 446–456.

[47] F. Samaria and A. Harter, "Parameterisation of a stochastic model for human face identification," in *Proc. 2nd IEEE Workshop on Applications of Computer Vision*, Sarasota FL, USA, Dec. 1994, pp. 138–142.

[48] K. Lee, J. Ho, and D. 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.

[49] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression database," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1615–1618, Dec. 2003.

[50] D. Cai, X. He, Y. Hu, J. Han, and T. S. Huang, "Learning a spatially smooth subspace for face recognition," in *IEEE Conf. Comput. Vis. and Pattern Recognit.*, Minneapolis, Minnesota, USA, Jun. 2007, pp. 138–142.

[51] "Four face databases in matlab format," http://http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html, accessed 2012-05-01.

[52] G. Rätsch, T. Onoda, and K.-R. Müller, "Soft margins for adaboost," *Mach. Learn.*, vol. 42, no. 3, pp. 287–320, Mar. 2001.

[53] C. E. Thomaz, D. F. Gillies, and R. Q. Feitosa, "A new covariance estimate for bayesian classifiers in biometric recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 2, pp. 214–223, Feb. 2004.

[54] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," *Psychometrika*, vol. 12, no. 2, pp. 153–157, Jun. 1947.

[55] 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.*, vol. 23, no. 3, pp. 526–534, Mar. 2012.

[56] Y. Aksu, D. J. Miller, G. Kesidis, and Q. X. Yang, "Margin-maximizing feature elimination methods for linear and nonlinear kernel-based discriminant functions," *IEEE Trans. Neural Netw.*, vol. 21, no. 5, pp. 701–717, May 2010.

[57] F. Song, D. Mei, and H. Li, "Feature selection based on linear discriminant analysis," in *IEEE Int. Conf. Intell. Syst. Design and Eng. Appl.*, vol. 1, Changsha, China, Oct. 2010, pp. 746–749.

[58] S. Huh and D. Lee, "Linear discriminant analysis for signatures," *IEEE Trans. Neural Netw.*, vol. 21, no. 12, pp. 1990–1996, Dec. 2010.