Projected Gradients for Subclass Discriminant Nonnegative Subspace Learning

Symeon Nikitidis, Anastasios Tefas, and Ioannis Pitas

***Abstract*—Current discriminant nonnegative matrix factorization (NMF) methods either do not guarantee convergence to a stationary limit point or assume a compact data distribution inside classes, thus ignoring intra class variance in extracting discriminant data samples representations. To address both limitations, we regard that data inside each class has a multimodal distribution, forming various subclasses and perform optimization using a projected gradients framework to ensure limit point stationarity. The proposed method combines appropriate clustering-based discriminant criteria in the NMF decomposition cost function, in order to find discriminant projections that enhance class separability in the reduced dimensional projection space, thus improving classification performance. The developed algorithms have been applied to facial expression, face and object recognition, and experimental results verified that they successfully identified discriminant parts, thus enhancing recognition performance.**

***Index Terms*—Face recognition, facial expression recognition, nonnegative matrix factorization, object recognition, subclass discriminant analysis.**

I. Introduction

IT IS COMMON knowledge that the spatial image dimensionality is much higher than that exploited by many image analysis applications. This fact necessitates seeking efficient dimensionality reduction methods for appropriate image feature extraction, which not only alleviate computational complexity but also boost performance of succeeding processing algorithms. One such popular category of methods is the subspace image representation algorithms which aim to discover the latent image features by projecting linearly or nonlinearly an image to a low dimensional subspace, where a certain criterion is optimized.

Nonnegative matrix factorization (NMF) [1] is such a popular algorithm widely used in image processing. It is an unsupervised data matrix decomposition method that requires both the matrix being decomposed and the derived factors to contain nonnegative elements. The nonnegativity constraint imposed by NMF on both the latent variables and the observations is especially meaningful when we operate on image data since the underlying features are naturally nonnegative. Moreover, in this case the semantic interpretability of the nonnegative subspace learning is enhanced, since this conforms nicely to identifying appropriate basic elements which are added to reconstruct the original image. This nonnegativity constraint distinguishes NMF from many other traditional dimensionality reduction methods, such as principal component analysis (PCA) [2], independent component analysis (ICA) [3], [4] or singular value decomposition (SVD) [5].

One of the most useful properties of NMF-based methods is that they usually produce a sparse representation of the decomposed data. Sparse coding corresponds to a data representation using few basic elements that are spatially distributed and ideally nonoverlapping. However, because the sparseness achieved by the original NMF is somewhat of a side-effect rather than a goal, caused by the imposed nonnegativity constraints, different studies have attempted to control the degree to which the derived representation is sparse. Toward this direction, Hoyer [6] incorporated the notion of sparsity into the standard NMF decomposition function so as the sparseness of the representation can be better controlled, while Li *et al*. [7] introduced localization constraints, leading to a parts-based representation.

Recently, numerous specialized NMF-based algorithms have been proposed and applied in various problems in diverse fields. These algorithms modify the NMF decomposition cost function, by incorporating additional penalty terms in order to fulfill specific requirements, arising in each application domain. In [8], projective NMF (PNMF) was introduced, which proved to generate a much sparser and near orthogonal projection matrix compared to the original NMF. An extension of NMF that is applicable on mixed sign data has been attempted in [9], where the nonnegativity constraint on the basis images has been relaxed, while the weights matrix remained positively constrained. Toward improving clustering performance, Cai *et al*. [10] proposed the graph regularized NMF (GNMF) that encodes the local data geometric structure considering a nearest neighbor graph in order to exploit local geometrical invariance between training samples when these are mapped from the initial data space to the projection subspace. Other approaches that exploit the data geometric structure in order to extract discriminative information have been also proposed in [11] and [12]. Another notable variant of NMF which retains the manifold structure of facial space,

Manuscript received June 26, 2013; revised December 13, 2013 and March 23, 2014; accepted March 27, 2014. Date of publication May 5, 2014; date of current version November 13, 2014. This work was supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under Grant Agreement 248434 (MOBISERV). This paper was recommended by Associate Editor D. Tao.

S. Nikitidis was with the Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki 54124, Greece. He is now with the Department of Computing, Imperial College London, London, SW7 2AZ, U.K. (e-mail: s.nikitidis@imperial.ac.uk).

I. Pitas and A. Tefas are with the Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki 54124, Greece (e-mail: tefas@aiia.csd.auth.gr; pitas@aiia.csd.auth.gr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2014.2317174

is the topology preserving NMF (TPNMF) [13] specialized for face representation and recognition.

Focusing on applications operating on facial image data, numerous specialized NMF variants have been proposed for face recognition [7], [13], [14], face verification [15], [16], and facial expression recognition [17], [18]. In these approaches, the entire facial image is considered as a feature vector and NMF aims to find projections that optimize a given criterion. The resulting projections are then used in order to project unknown test facial images to a lower dimensional subspace where the criteria under consideration are optimized. In order to model properly the nonlinearities that are present in most real life applications, polynomial NMF (PNMF) has been proposed in [19], which projects the original data into polynomial spaces of arbitrary degree. An extension of PNMF has been proposed in [20], that considers projection of the training data using arbitrary Mercer's kernels.

A supervised NMF learning method that aims to extract discriminant facial parts appropriate for face verification is the discriminant NMF (DNMF) algorithm [15]. DNMF incorporates additional terms inspired by linear discriminant analysis (LDA) [21] in the NMF factorization and achieves a more efficient decomposition of the provided data in their discriminant parts, thus enhancing separability between classes. However, the considered discriminant factor possesses two certain deficiencies inherited from the LDA optimality assumption. Firstly, it assumes that the data samples of each class are generated from underlying multivariate normal distributions of a common covariance matrix but with different means. Secondly, since this approach assumes that each class is represented by a single compact data cluster, the problem of nonlinearly separable classes cannot be treated efficiently. Unfortunately, in various real world applications, data distribution usually does not correspond to compact sets but data form various subclasses. This is common in face recognition due to various factors such as pose and illumination variations or in facial expression recognition, since there is no unique way that people form certain expressions [22]. Moreover, subclasses formation is also evident on data describing human activities, since certain actions can be performed by different bodily manifestations.

To overcome the aforementioned limitations, we relax the assumption that each class consists of a single compact data cluster and regard that it is composed of various subclasses, where each one is approximated by a Gaussian distribution. Consequently, we approximate the underlying distribution of each class as a mixture of Gaussians and apply criteria inspired by the clustering-based discriminant analysis (CDA) [22] to enhance discrimination between different classes. Moreover, we extend NMF reformulating its cost function by embedding appropriate discriminant constraints and propose a novel algorithm, called subclass discriminant NMF (SDNMF), which finds discriminant projections that enhance class separability in the reduced dimensional space, by imposing discriminant criteria that assume multimodality of the training data. To solve the SDNMF problem, we both consider multiplicative update rules and iterative projected gradients optimization algorithms in order to exploit their well established optimization properties [23]–[25] that ensure stationarity of the reached limit point. Finally, we derive the nonlinear counterpart of SDNMF that projects training data to high dimensional Hilbert spaces and propose a set of update rules that consider polynomial projection spaces of arbitrary degree.

In summary, the novel contributions of this paper are as follows.

1) An NMF-based algorithm called SDNMF that assumes a multimodal data distribution inside classes is proposed.
2) To solve SDNMF, novel update rules under two different optimization frameworks are proposed and their optimization properties and proof of convergence are exhibited.
3) The nonlinear counterpart of SDNMF algorithm that considers a polynomial projection space is demonstrated.
4) A thorough experimental study on various image recognition problems is performed, comparing the proposed methods with current state-of-the-art linear dimensionality reduction algorithms.

The rest of the paper is organized as follows. The linear and nonlinear NMF (NNMF) algorithms, as well as DNMF are reviewed in Section II. Section III introduces the CDA inspired discriminant criteria, the proposed SDNMF method and the developed update rules considering two different optimization strategies. Moreover, the nonlinear counterpart of SDNMF is also demonstrated. Section IV presents the conducted experimental study and verifies the efficiency of our algorithms for facial expression, face, and object recognition. Finally, concluding remarks are drawn in Section V. A preliminary version of this paper can be found in [26] and [27].

II. Linear and Nonlinear NMF and Its Discriminant Variant

Next, we briefly present the linear and nonlinear NMF decomposition concept and also review DNMF algorithm. In the following, without losing generality, we shall assume that the decomposed data are images, although, the techniques that will be described can be applied to any kind of nonnegative data.

A. NMF Basics

The basic idea of NMF is to approximate an image by a linear combination of elements, the so-called basis images, that correspond to image parts. Let $\mathcal{I}$ be an image database comprised of L images belonging to n different classes and $\mathbf{X} \in R_+^{F\times L}$ be the data matrix whose columns are F-dimensional feature vectors obtained by scanning row-wise each image in the database. NMF considers factorizations of the form

$$\mathbf{X} \approx \mathbf{ZH} \tag{1}$$

where $\mathbf{Z} \in R_+^{F\times M}$ (with $M \ll F$) is a matrix containing the basis images, while matrix $\mathbf{H} \in R_+^{M\times L}$ contains the coefficients of the linear combinations of the basis images required to reconstruct each original image in the database. Thus, after the NMF decomposition the j-th image $\mathbf{x}_j$ can be approximated by $\mathbf{x}_j \approx \mathbf{Zh}_j$, where $\mathbf{h}_j$ denotes the j-th weight column of matrix $\mathbf{H}$.

To measure the cost of the decomposition $\mathcal{O}(\mathbf{X}||\mathbf{ZH})$ in (1), commonly the matrix Frobenius norm square is used which computes the sum of the squared Euclidean distances between all original images in the database and their respective reconstructed versions

$$\mathcal{O}(\mathbf{X}||\mathbf{ZH}) \triangleq ||\mathbf{X}-\mathbf{ZH}||_F^2 = \sum_{j=1}^{L}\sum_{i=1}^{F}\left(x_{i,j}-[\mathbf{ZH}]_{i,j}\right)^2 \quad (2)$$

where $||.||_F$ is the Frobenius norm. Hence, NMF algorithm factorizes the data matrix $\mathbf{X}$ into $\mathbf{ZH}$, by solving the following constrained optimization problem:

$$\begin{aligned} &\min_{\mathbf{Z},\mathbf{H}} \mathcal{O}(\mathbf{X}||\mathbf{ZH}) \quad (3)\\ &\text{subject to:} \quad z_{i,k}\geq 0, \quad h_{k,j}\geq 0, \quad \forall i,j,k. \end{aligned}$$

Using an appropriately designed auxiliary function, it has been shown in [28] that the following multiplicative rules update $h_{k,j}$ and $z_{i,k}$, resulting in the desired factors, while guaranteeing a non-increasing behavior of the cost function:

$$h_{k,j}^{(t)} = h_{k,j}^{(t-1)}\frac{[\mathbf{Z}^{(t-1)^T}\mathbf{X}]_{k,j}}{[\mathbf{Z}^{(t-1)^T}\mathbf{Z}^{(t-1)}\mathbf{H}^{(t-1)}]_{k,j}} \quad (4)$$

$$z_{i,k}^{(t)} = z_{i,k}^{(t-1)}\frac{[\mathbf{X}\mathbf{H}^{(t)^T}]_{i,k}}{[\mathbf{Z}^{(t-1)}\mathbf{H}^{(t)}\mathbf{H}^{(t)^T}]_{i,k}}. \quad (5)$$

B. Nonlinear NMF

The problem of NNMF can be summarized as follows: find a set of nonnegative weights and nonnegative, nonlinear basis vectors such that the nonnegative nonlinearly mapped training data can be approximated as a linear combination of the learned nonnegative nonlinearly mapped basis vectors. This can be formulated as follows. Let $\phi(\mathbf{x}_i):R_+^F\rightarrow\mathcal{H}$ be a nonlinear mapping function that projects the input image $\mathbf{x}_i$ to an arbitrary dimensional Hilbert space $\mathcal{H}$ where NNMF considers the following factorization:

$$\mathbf{X}^{\phi}\approx\mathbf{Z}^{\phi}\mathbf{H} \quad (6)$$

where $\mathbf{X}^{\phi}=[\phi(\mathbf{x}_i),\ldots,\phi(\mathbf{x}_L)]$ $\mathbf{Z}^{\phi}=[\phi(\mathbf{z}_1),\ldots,\phi(\mathbf{z}_M)]$ and $\mathbf{H}\in R_+^{M\times L}$ contains the coefficients of the linear combinations of the mapped basis vectors $\phi(\mathbf{z}_j)$ required to perform the approximation. The approximation error can be similarly measured using the Frobenius norm square

$$\begin{aligned}\mathcal{O}\left(\mathbf{X}^{\phi}||\mathbf{Z}^{\phi}\mathbf{H}\right) &\triangleq \frac{1}{2}\sum_{j=1}^{L}||\phi(\mathbf{x}_j)-\sum_{k=1}^{M}h_{k,j}\phi(\mathbf{z}_k)||_F^2\\ &= \frac{1}{2}\sum_{j=1}^{L}\Big([\mathbf{K}_{x,x}]_{j,j}-2\sum_{k=1}^{M}h_{k,j}[\mathbf{K}_{z,x}]_{k,j}\\ &\quad+\sum_{k=1}^{M}\sum_{l=1}^{M}h_{k,j}h_{l,j}[\mathbf{K}_{z,z}]_{l,k}\Big) \quad (7)\end{aligned}$$

where the kernel matrices are defined as

$$\begin{aligned}&\left[\mathbf{K}_{x,x}\right]_{i,j}=\phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j), \ \left[\mathbf{K}_{z,z}\right]_{i,j}=\phi(\mathbf{z}_i)^T\phi(\mathbf{z}_j)\\ &\left[\mathbf{K}_{z,x}\right]_{i,j}=\phi(\mathbf{z}_i)^T\phi(\mathbf{x}_j), \ \mathbf{K}_{x,z}=\mathbf{K}_{z,x}^T. \quad (8)\end{aligned}$$

Thus, NNMF solves the following optimization problem:

$$\begin{aligned} &\min_{\mathbf{Z},\mathbf{H}} \mathcal{O}_{\phi}(\mathbf{X}^{\phi}||\mathbf{Z}^{\phi}\mathbf{H}) \quad (9)\\ &\text{subject to:} \quad z_{i,k}\geq 0 \quad h_{k,j}\geq 0 \end{aligned}$$

where $i=1,\ldots,F$, $j=1,\ldots,L$ and $k=1,\ldots,M$. In [19], polynomial kernels of the form $k(\mathbf{x}_i,\mathbf{x}_j)=(\mathbf{x}_i^T\mathbf{x}_j)^d$ were considered, where d denotes the polynomial degree and the respective solution was found using appropriate auxiliary functions of the actually minimized cost function for both variables $\mathbf{Z}$ and $\mathbf{H}$. Thus, the following multiplicative update rules were proposed for minimizing (7):

$$\mathbf{H}^{(t)}=\mathbf{H}^{(t-1)}\odot\frac{\mathbf{K}_{x,z}^{(t-1)}}{\left(\mathbf{K}_{z,z}^{(t-1)}\mathbf{H}^{(t-1)}\right)} \quad (10)$$

$$\hat{\mathbf{Z}}^{(t)}=\mathbf{Z}^{(t-1)}\odot\frac{\mathbf{X}\acute{\mathbf{K}}_{x,z}^{(t-1)}}{\mathbf{Z}^{(t-1)}\boldsymbol{\Omega}\acute{\mathbf{K}}_{z,z}^{(t-1)}}, \quad \mathbf{Z}^{(t)}=\frac{\hat{\mathbf{Z}}^{(t)}}{\mathbf{S}} \quad (11)$$

where $\boldsymbol{\Omega}$ is a diagonal matrix, with $[\boldsymbol{\Omega}]_{j,j}=\sum_{k=1}^{M}h_{k,j}$ and $\mathbf{S}$ is a normalization matrix, such that the columns of $\mathbf{Z}^{(t)}$ sum up to one. Matrices $\acute{\mathbf{K}}_{x,z}$ and $\acute{\mathbf{K}}_{z,z}$ contain parts of the first order derivatives with respect to $z_{i,k}$ of the polynomial kernels and are defined as $[\acute{\mathbf{K}}_{x,z}]_{i,j}=d(\mathbf{x}_i^T\mathbf{z}_j)^{d-1}$ and $[\acute{\mathbf{K}}_{z,z}]_{i,j}=d(\mathbf{z}_i^T\mathbf{z}_j)^{d-1}$. Operators $\odot$ and / denote element-wise multiplication and division of matrices, respectively.

C. Discriminant NMF

DNMF [15] is an attempt to introduce discriminant constraints in the NMF decomposition cost function. It exploits the well known Fisher discriminant criterion which attempts to find a transformation matrix $\boldsymbol{\Psi}$ that maximizes the ratio defined by the traces of the between and within class scatter matrices $\acute{\mathbf{S}}_b=\boldsymbol{\Psi}^T\mathbf{S}_b\boldsymbol{\Psi}$ and $\acute{\mathbf{S}}_w=\boldsymbol{\Psi}^T\mathbf{S}_w\boldsymbol{\Psi}$ evaluated over the projected data. DNMF cost function incorporates a similar discriminant factor, requiring the dispersion of the projected samples that belong to the same class around their corresponding mean to be as small as possible, while at the same time the scatter of the mean vectors of all classes around their global mean to be as large as possible. Consequently, DNMF minimizes the following cost function:

$$\mathcal{O}_{DNMF}(\mathbf{X}||\mathbf{ZH})=\mathcal{O}_{KL}(\mathbf{X}||\mathbf{ZH})+\alpha\mathrm{Tr}[\acute{\mathbf{S}}_w]-\beta\mathrm{Tr}[\acute{\mathbf{S}}_b] \quad (12)$$

where $\mathcal{O}_{KL}(\mathbf{X}||\mathbf{ZH})$ measures the reconstruction error using the Kullback–Leibler (KL) divergence metric, Tr[.] is the matrix trace operator and α, β are positive constants.

III. Subclass Discriminant Nonnegative Matrix Factorization

In this section, we first present the subclass-based discriminant criteria and demonstrate how these are incorporated in the NMF decomposition cost function resulting in the SDNMF problem. Next, we derive the proposed update rules considering two different optimization strategies and also present SDNMF nonlinear counterpart.

A. Subclass Discriminant Analysis

Similar to LDA, CDA seeks to determine a transformation matrix Ψ that enhances classes discrimination in the projection subspace. To do so, CDA assumes a multimodal data distribution inside classes, where each class is composed of various subclasses and attempts to enhance classes discrimination by minimizing the scatter within every subclass, while well separating subclasses from each other class.

To formulate the CDA criteria for the n-class image database $\mathcal{I}$, let us denote the number of subclasses composing the r-th class by C_r, the total number of formed subclasses in the database by $C = \sum_i^n C_i$ and the number of images belonging to the θ-th subclass of the r-th class by $N_{r,\theta}$. Let us also define the mean vector for the θ-th cluster of the r-th class by $\boldsymbol{\mu}^{r,\theta} = [\mu_1^{r,\theta} \dots \mu_F^{r,\theta}]^T$, which is evaluated over the $N_{r,\theta}$ images, while vector $\mathbf{x}_\rho^{r,\theta} = [x_{\rho,1}^{r,\theta} \dots x_{\rho,F}^{r,\theta}]^T$ corresponds to the feature vector of the ρ-th image belonging to the θ-th cluster of the r-th class. Using the above notations we can define the within subclass scatter matrix $\mathbf{S}_w^{CDA}$ as

$$\mathbf{S}_w^{CDA} = \sum_{r=1}^{n} \sum_{\theta=1}^{C_r} \sum_{\rho=1}^{N_{r,\theta}} \left(\mathbf{x}_\rho^{r,\theta} - \boldsymbol{\mu}^{r,\theta}\right)\left(\mathbf{x}_\rho^{r,\theta} - \boldsymbol{\mu}^{r,\theta}\right)^T \tag{13}$$

and the between subclass scatter matrix $\mathbf{S}_b^{CDA}$ as

$$\mathbf{S}_b^{CDA} = \sum_{i=1}^{n} \sum_{r,r\neq i}^{n} \sum_{j=1}^{C_i} \sum_{\theta=1}^{C_r} \left(\boldsymbol{\mu}^{i,j} - \boldsymbol{\mu}^{r,\theta}\right)\left(\boldsymbol{\mu}^{i,j} - \boldsymbol{\mu}^{r,\theta}\right)^T. \tag{14}$$

Considering that the columns of matrix $\mathbf{H}$ contain the projected M-dimensional feature vectors and in order to facilitate our subsequent analysis using more compact equation forms, we express the CDA scatter matrices in a graph Laplacian form

$$\Sigma_w \triangleq \sum_{r=1}^{n} \sum_{\theta=1}^{C_r} \sum_{j=1}^{N_{r,\theta}} \left(\mathbf{h}_j - \boldsymbol{\mu}^{r,\theta}\right)\left(\mathbf{h}_j - \boldsymbol{\mu}^{r,\theta}\right)^T = \mathbf{H}\mathbf{L}_w\mathbf{H}^T \tag{15}$$

and

$$\Sigma_b \triangleq \sum_{i=1}^{n} \sum_{r,r\neq i}^{n} \sum_{j=1}^{C_i} \sum_{\theta=1}^{C_r} \left(\boldsymbol{\mu}^{i,j} - \boldsymbol{\mu}^{r,\theta}\right)\left(\boldsymbol{\mu}^{i,j} - \boldsymbol{\mu}^{r,\theta}\right)^T = \mathbf{H}\mathbf{L}_b\mathbf{H}^T \tag{16}$$

where $\mathbf{L}_w$ and $\mathbf{L}_b$ are $L \times L$ symmetric positive semidefinite matrices defined as

$$\mathbf{L}_w \triangleq \mathbf{I}_L - \sum_{r=1}^{n} \sum_{\theta=1}^{C_r} \left(\frac{1}{N_{r,\theta}} \mathbf{e}_{r,\theta}^T \mathbf{e}_{r,\theta}\right) \tag{17}$$

$$\mathbf{L}_b \triangleq 2\Bigg(\sum_{r=1}^{n} \sum_{\theta=1}^{C_r} \frac{C - C_r}{N_{r,\theta}^2} \mathbf{e}_{r,\theta}^T \mathbf{e}_{r,\theta} - \mathrm{diag}(\mathbf{e}) \times \left[\mathbf{1} - \sum_{r=1}^{C_r} \mathbf{e}_r^T \mathbf{e}_r\right] \mathrm{diag}(\mathbf{e}) \Bigg). \tag{18}$$

Here diag($\mathbf{e}$) denotes a function that converts vector $\mathbf{e}$ into a diagonal matrix, $\mathbf{I}_L$ is an $L \times L$ identity matrix, $\mathbf{1}$ is an $L \times L$ matrix of ones, while $\mathbf{e}_{r,\theta}$, $\mathbf{e}_r$ and $\mathbf{e}$ are L-dimensional vectors whose i-th element is defined as

$$[\mathbf{e}_{r,\theta}]_i = \begin{cases} 1, & \text{if } \mathbf{x}_i \in \theta\text{-th cluster of the } r\text{-th class} \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

$$[\mathbf{e}_r]_i = \begin{cases} 1, & \text{if } \mathbf{x}_i \in r\text{-th class} \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

$$[\mathbf{e}]_i = \frac{1}{\text{Cardinality of sample } \mathbf{x}_i \text{ cluster}}. \tag{21}$$

The trace of the within subclass scatter matrix Σ_w can be used as an appropriate indicator of the samples dispersion inside subclasses. Minimizing $\mathrm{Tr}[\Sigma_w]$ increases concentration of samples around their subclass mean. Similarly, $\mathrm{Tr}[\Sigma_b]$ indicates the dispersion of the mean vectors between all subclasses that belong to different classes. Thus, maximizing $\mathrm{Tr}[\Sigma_b]$ enhances discrimination between subclasses of different classes.

B. SDNMF Objective Function and Its Multiplicative Update Rules

Since we desire in the projection subspace to simultaneously minimize $\mathrm{Tr}[\Sigma_w]$ and maximize $\mathrm{Tr}[\Sigma_b]$, the cost function of the SDNMF algorithm is formulated as follows:

$$\mathcal{O}_{SDNMF}(\mathbf{X}||\mathbf{ZH}) \triangleq \frac{1}{2}||\mathbf{X} - \mathbf{ZH}||_F^2 + \frac{\alpha}{2}\mathrm{Tr}[\mathbf{H}\mathbf{L}_w\mathbf{H}^T] - \frac{\beta}{2}\mathrm{Tr}[\mathbf{H}\mathbf{L}_b\mathbf{H}^T] \tag{22}$$

where α and β are positive constants. Equivalently, (22) can be written in a matrix trace form as follows:

$$\mathcal{O}_{SDNMF}(\mathbf{X}||\mathbf{ZH}) = \frac{1}{2}\mathrm{Tr}[\mathbf{X}\mathbf{X}^T] - \mathrm{Tr}[\mathbf{Z}\mathbf{H}\mathbf{X}^T] + \frac{1}{2}\mathrm{Tr}[\mathbf{Z}\mathbf{H}\mathbf{H}^T\mathbf{Z}^T] + \frac{\alpha}{2}\mathrm{Tr}[\mathbf{H}\mathbf{L}_w\mathbf{H}^T] - \frac{\beta}{2}\mathrm{Tr}[\mathbf{H}\mathbf{L}_b\mathbf{H}^T] \tag{23}$$

where we have applied properties $\mathrm{Tr}[\mathbf{AB}] = \mathrm{Tr}[\mathbf{BA}]$ $\mathrm{Tr}[\mathbf{A}] = \mathrm{Tr}[\mathbf{A}^T]$ and $||\mathbf{A}||_F^2 = \mathrm{Tr}[\mathbf{A}\mathbf{A}^T]$.

Consequently, the minimization problem of SDNMF is formulated as

$$\min_{\mathbf{Z},\mathbf{H}} \mathcal{O}_{SDNMF}(\mathbf{X}||\mathbf{ZH}) \quad \text{subject to:} \quad z_{i,k} \geq 0 \quad h_{k,j} \geq 0, \quad \forall i,j,k. \tag{24}$$

which requires the minimization of (23) subject to the nonnegativity constraints applied on the elements of both factors $\mathbf{H}$ and $\mathbf{Z}$. To solve (24), we introduce Lagrange multipliers $\boldsymbol{\phi} \in R_+^{F\times M} = [\phi_{i,k}]$ and $\boldsymbol{\psi} \in R_+^{M\times L} = [\psi_{k,j}]$ each associated with one of the nonnegativity constraints $z_{i,k} \geq 0$, $h_{k,j} \geq 0$, respectively. Consequently, we formulate the Lagrangian function $\mathcal{L}$ as follows:

$$\mathcal{L} = \frac{1}{2}\mathrm{Tr}[\mathbf{X}\mathbf{X}^T] - \mathrm{Tr}[\mathbf{Z}\mathbf{H}\mathbf{X}^T] + \frac{1}{2}\mathrm{Tr}[\mathbf{Z}\mathbf{H}\mathbf{H}^T\mathbf{Z}^T] + \mathrm{Tr}[\boldsymbol{\psi}\mathbf{H}^T] + \frac{\alpha}{2}\mathrm{Tr}[\mathbf{H}\mathbf{L}_w\mathbf{H}^T] - \frac{\beta}{2}\mathrm{Tr}[\mathbf{H}\mathbf{L}_b\mathbf{H}^T] + \mathrm{Tr}[\boldsymbol{\phi}\mathbf{Z}^T]. \tag{25}$$

To minimize $\mathcal{L}$, we first obtain its partial derivatives with respect to $z_{i,k}$ and $h_{k,j}$ and set them equal to zero

$$\frac{\partial \mathcal{L}}{\partial h_{k,j}} = [\mathbf{Z}^T\mathbf{Z}\mathbf{H}]_{k,j} - [\mathbf{Z}^T\mathbf{X}]_{k,j} + \alpha[\mathbf{H}\mathbf{L}_w]_{k,j} - \beta[\mathbf{H}\mathbf{L}_b]_{k,j} + \psi_{k,j} = 0 \quad (26)$$

$$\frac{\partial \mathcal{L}}{\partial z_{i,k}} = [\mathbf{Z}\mathbf{H}\mathbf{H}^T]_{i,k} - [\mathbf{X}\mathbf{H}^T]_{i,k} + \phi_{i,k} = 0. \quad (27)$$

According to KKT conditions [29] $\phi_{i,k}z_{i,k} = 0$ and also $\psi_{k,j}h_{k,j} = 0$. Consequently, multiplying (26) with $h_{k,j}$ and (27) with $z_{i,k}$ we obtain the following equalities:

$$\left(\frac{\partial \mathcal{L}}{\partial h_{k,j}}\right) h_{k,j} = [\mathbf{Z}^T\mathbf{Z}\mathbf{H}]_{k,j}h_{k,j} - [\mathbf{Z}^T\mathbf{X}]_{k,j}h_{k,j} + \alpha[\mathbf{H}\mathbf{L}_w]_{k,j}h_{k,j} - \beta[\mathbf{H}\mathbf{L}_b]_{k,j}h_{k,j} = 0 \quad (28)$$

$$\left(\frac{\partial \mathcal{L}}{\partial z_{i,k}}\right) z_{i,k} = [\mathbf{Z}\mathbf{H}\mathbf{H}^T]_{i,k}z_{i,k} - [\mathbf{X}\mathbf{H}^T]_{i,k}z_{i,k} = 0. \quad (29)$$

The added discriminant factors in the SDNMF cost function are totally independent from the basis image matrix $\mathbf{Z}$. Consequently, keeping variable $\mathbf{H}$ fixed and optimizing for $\mathbf{Z}$ results to the same optimization problem described in [28] and to the update formulas in (5). This can be also verified by solving (29) for $z_{i,k}$. Thus, we can recall the convergence proof of conventional NMF in [28] to show that (23) is non-increasing under the update rule in (5). Solving (28) for $h_{k,j}$ we derive the proposed multiplicative update rule shown in (30), at the bottom of the page. The detailed proof regarding the non-increasing behavior of (23) under the proposed update rule in (30) is derived using an auxiliary upper bounding function which can be found in the Appendix A. Similar convergence proofs are widely used for a variety of optimization problems solved using multiplicative update rules [30], [31].

It should be noted that as in every NMF-based optimization problem, the objective function in (23) is convex either in $\mathbf{Z}$ or $\mathbf{H}$, but non-convex in both variables. Therefore, the proposed iterative optimization algorithm reaches a locally optimal solution which is non-unique and is usually sensitive to the initialization point. Various initialization strategies have been proposed in the literature [32], [33], however, their efficacy is both data and application dependant, since the additional imposed constraints in the NMF decomposition cost function also affect the starting factors suitability. Lee and Seung [1] exploited the random seeding approach which is computationally efficient and has been also adopted in this paper.

The optimization process successively updates variable $\mathbf{Z}$ and $\mathbf{H}$ until a stopping criterion is invoked. In this paper, we terminate the optimization process when the cost function improvement between two successive iterations is less than 10^{-3}, since we have strong evidence of algorithms convergence. Other similar stopping criteria based on monitoring the objective function improvement have been proposed in the literature [25]. Finally, in order to extract the discriminant representation of an unknown test sample $\mathbf{x}_j$ we use the pseudo-inverse $\mathbf{Z}^\dagger = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T$ as $\acute{\mathbf{x}}_j = \mathbf{Z}^\dagger\mathbf{x}_j$. The iterative optimization process for the SDNMF problem is summarized in Algorithm 1.

Algorithm 1 Algorithm Outline for SDNMF Optimization

1: **Input:** Nonnegative data matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L]$ along with the class label and cluster origin $\{y_i, c_i\}$ associated with each training facial image $\mathbf{x}_i$ $i = 1, \ldots, L$.
2: **Output:** The basis images matrix $\mathbf{Z} \in R_+^{F\times M}$ and the weights matrix $\mathbf{H} \in R_+^{M\times L}$.
3: **Initialize:** $\mathbf{Z}^{(0)}$, $\mathbf{H}^{(0)}$ and $t = 1$.
4: **repeat**
5: **Update** $\mathbf{H}^{(t)}$ given $\mathbf{Z}^{(t-1)}$ using (30).
6: **Update** $\mathbf{Z}^{(t)}$ given $\mathbf{H}^{(t)}$ using (5).
7: $t = t + 1$.
8: **until** $|\mathcal{O}_{SDNMF}(\mathbf{X}||\mathbf{Z}^{(t)}\mathbf{H}^{(t)}) - \mathcal{O}_{SDNMF}(\mathbf{X}||\mathbf{Z}^{(t-1)}\mathbf{H}^{(t-1)})| \leq 10^{-3}$

C. Dividing Classes Into Subclasses

Regarding the optimal division of each class into subclasses, various criteria have been proposed in the literature [34], [35]. In our implementation, we have considered the nearest-neighbor (NN)-based clustering algorithm presented in [34] which is a good compromise between computation speed and clustering accuracy. Moreover, as it has been shown in [34] various other clustering methods can be used but they do not affect the overall classification performance significantly. This can be attributed to the fact that only first and second order statistics of each cluster are used in the optimization criteria and, thus, precise clustering is not crucial, as long as the location and dispersion of each cluster is robustly estimated.

According to NN clustering, we first construct a sorted set $\{\mathbf{x}_{r,1}, \ldots, \mathbf{x}_{r,N_r}\}$ for every r-th class with its N_r training samples arranged as follows: samples $\mathbf{x}_{r,1}$ and $\mathbf{x}_{r,N_r}$ are the two most distant feature vectors of the r-th class in the initial high dimensional image space (i.e., those that maximize the Euclidean distance $\text{argmax}_{\mathbf{x}_i,\mathbf{x}_j} ||\mathbf{x}_i - \mathbf{x}_j||_2$). The rest of the samples are then ordered, so that $\mathbf{x}_{r,2}$ is the sample closest to $\mathbf{x}_{r,1}$, while $\mathbf{x}_{r,N_r-1}$ is the sample closest to $\mathbf{x}_{r,N_r}$. This procedure results in an ordered set, where the sample ranked in the j-th position is the $(j-1)$-th closest sample to $\mathbf{x}_{r,1}$, and at the same time, the $(N_r - j)$-th more distant sample to the other extremum $\mathbf{x}_{r,N_r}$. Subsequently, we divide data samples belonging to the r-th class into C_r subclasses, by partitioning the ordered set into C_r equally sized parts.

D. Projected Gradients Subclass Discriminant Nonnegative Matrix Factorization (PGSDNMF)

The derived multiplicative update rules for the evaluation of the optimal factors lack of convergence results since they

$$h_{k,j}^{(t)} = h_{k,j}^{(t-1)} \frac{[\mathbf{Z}^{(t-1)^T}\mathbf{X}]_{k,j} + \beta[\mathbf{H}^{(t-1)}\sum_{r=1}^{n}\sum_{\theta=1}^{C_r}\frac{C-C_r}{N_{r,\theta}^2}\mathbf{e}_{r,\theta}^T\mathbf{e}_{r,\theta}]_{k,j}}{[\mathbf{Z}^{(t-1)^T}\mathbf{Z}^{(t-1)}\mathbf{H}^{(t-1)}]_{k,j} + \alpha[\mathbf{H}^{(t-1)}\mathbf{L}_w]_{k,j} + \beta\left[\mathbf{H}^{(t-1)}\text{diag}(\mathbf{e})\left(\mathbf{1} - \sum_{r=1}^{C_r}\mathbf{e}_r^T\mathbf{e}_r\right)\text{diag}(\mathbf{e})\right]_{k,j}} \quad (30)$$

only guarantee a non-increasing behavior of the cost function in (23) and do not ensure that optimization converges to a limit point that is also stationary [23], [24]. Moreover, as it has been shown in [24], update rules derived using projected gradients attain faster convergence compared to their multiplicative counterparts. In order to exploit these merits we employ projected gradients for SDNMF optimization. To do so, we formulate two subproblems $\mathcal{O}_1(\mathbf{Z})$ and $\mathcal{O}_2(\mathbf{H})$ from (23), by keeping either $\mathbf{H}$ or $\mathbf{Z}$ fixed and performing optimization for the other variable

$$\min_{\mathbf{Z}} \mathcal{O}_1(\mathbf{Z}) \quad \text{subject to:} \quad z_{i,k} \geq 0, \quad \forall i,k \tag{31}$$

$$\min_{\mathbf{H}} \mathcal{O}_2(\mathbf{H}) \quad \text{subject to:} \quad h_{k,j} \geq 0, \quad \forall k,j. \tag{32}$$

1) Optimization of $\mathbf{Z}$ *Solving the Subproblem (31):* The performed optimization is an iterative steepest descent process that at a given iteration round t the following update rule is applied:

$$\mathbf{Z}^{(t)} = P[\mathbf{Z}^{(t-1)} - \alpha_t \nabla \mathcal{O}_1(\mathbf{Z}^{(t-1)})] \tag{33}$$

where operator $P[.] = \max[.,0]$ guarantees that no negative values can be assigned to the updated elements of matrix $\mathbf{Z}$ and α_t is the learning step parameter for the t-th iteration. To set the learning step parameter α_t, we use the Armijo rule as in [24]. According to this strategy the learning step is computed as $\alpha_t = \beta^{g_t}$, where g_t is the first nonnegative integer value found, such that the following inequality is satisfied:

$$\mathcal{O}_1(\mathbf{Z}^{(t)}) - \mathcal{O}_1(\mathbf{Z}^{(t-1)}) \leq \sigma \langle \nabla \mathcal{O}_1(\mathbf{Z}^{(t-1)}), \mathbf{Z}^{(t)} - \mathbf{Z}^{(t-1)} \rangle \tag{34}$$

where operator $\langle .,. \rangle$ is the Frobenius inner product, while parameters β and σ have been set to $\beta = 0.1$ and $\sigma = 0.01$ which is an efficient parameter selection, as has been verified in other studies [24], [36].

Since $\mathcal{O}_1(\mathbf{Z})$ is quadratic in terms of $\mathbf{Z}$ it can be expanded near $\mathbf{Z}^{(t-1)}$ as follows:

$$\begin{aligned}\mathcal{O}_1\left(\mathbf{Z}^{(t)}\right) = \mathcal{O}_1\left(\mathbf{Z}^{(t-1)}\right) + \left(\mathbf{Z}^{(t)} - \mathbf{Z}^{(t-1)}\right)^T \nabla \mathcal{O}_1\left(\mathbf{Z}^{(t-1)}\right) \\ + \tfrac{1}{2}\left(\mathbf{Z}^{(t)} - \mathbf{Z}^{(t-1)}\right)^T \nabla^2 \mathcal{O}_1\left(\mathbf{Z}^{(t-1)}\right)\left(\mathbf{Z}^{(t)} - \mathbf{Z}^{(t-1)}\right).\end{aligned} \tag{35}$$

By replacing (35) into (34), we derive the actually checked condition, which is less computationally expensive than (34)

$$\begin{aligned}(1-\sigma)\left\langle \nabla \mathcal{O}_1(\mathbf{Z}^{(t-1)})\mathbf{Z}^{(t)} - \mathbf{Z}^{(t-1)} \right\rangle \\ + \frac{1}{2}\left\langle \mathbf{Z}^{(t)} - \mathbf{Z}^{(t-1)}, \nabla^2 \mathcal{O}_1(\mathbf{Z}^{(t-1)})(\mathbf{Z}^{(t)} - \mathbf{Z}^{(t-1)}) \right\rangle \leq 0.\end{aligned} \tag{36}$$

By iterating the update rule in (33), a sequence of minimizers $\{\mathbf{Z}^{(t)}\}_{t=1}^{\infty}$ of $\mathcal{O}_1(\mathbf{Z})$ is generated and according to Bertsekas [37], it is guaranteed that a stationary point is found among its limit points. Thus, in order to verify if the currently reached limit point is stationary or not, we examine whether the following condition is satisfied:

$$||\nabla^P \mathcal{O}_1(\mathbf{Z}^{(t)})||_F \leq e_{\mathbf{Z}} ||\nabla^P \mathcal{O}_1(\mathbf{Z}^{(1)})||_F \tag{37}$$

where $\nabla^P \mathcal{O}_1(\mathbf{Z}^{(t)})$ is the projected gradient of $\mathcal{O}_1(\mathbf{Z}^{(t)})$, with respect to $\mathbf{Z}$, with its (i,k)-th element defined as

$$[\nabla^P \mathcal{O}_1(\mathbf{Z}^{(t)})]_{i,k} = \begin{cases} [\nabla \mathcal{O}_1(\mathbf{Z}^{(t)})]_{i,k}, & \text{if } z_{i,k} > 0 \\ \min\left(0, [\nabla \mathcal{O}_1(\mathbf{Z}^{(t)})]_{i,k}\right), & \text{if } z_{i,k} = 0 \end{cases} \tag{38}$$

and $e_{\mathbf{Z}}$ is a predefined stopping tolerance set to $e_{\mathbf{Z}} = 10^{-3}$.

2) Optimization of $\mathbf{H}$ *Solving the Subproblem (32):* In order to find a stationary limit point for $\mathcal{O}_2(\mathbf{H})$, a similar procedure is applied. Initially, the learning step parameter α_t is determined and the weights matrix $\mathbf{H}$ is updated as follows:

$$\mathbf{H}^{(t)} = P[\mathbf{H}^{(t-1)} - \alpha_t \nabla \mathcal{O}_2(\mathbf{H}^{(t-1)})] \tag{39}$$

until the function $\mathcal{O}_2(\mathbf{H})$ is sufficiently decreased and the following inequality resulting by performing the expansion near $\mathbf{H}^{(t-1)}$ considering up to quadratic terms holds:

$$\begin{aligned}(1-\sigma)\left\langle \nabla \mathcal{O}_2(\mathbf{H}^{(t-1)}), \mathbf{H}^{(t)} - \mathbf{H}^{(t-1)} \right\rangle \\ + \tfrac{1}{2}\left\langle \mathbf{H}^{(t)} - \mathbf{H}^{(t-1)}, \nabla^2 \mathcal{O}_2(\mathbf{H}^{(t-1)})(\mathbf{H}^{(t)} - \mathbf{H}^{(t-1)}) \right\rangle \leq 0.\end{aligned} \tag{40}$$

The update procedure is repeated, until the limit point of the sequence $\{\mathbf{H}^{(t)}\}_{t=1}^{\infty}$ becomes stationary which is similarly determined to (37).

The minimization of subproblems in (31) and (32) involves the calculation of the first and second order gradients of the two optimized functions $\mathcal{O}_1(\mathbf{Z})$ and $\mathcal{O}_2(\mathbf{H})$. Using the formulation of the subclass scatter matrices provided in (15) and (16), the partial derivatives are evaluated as follows:

$$\nabla \mathcal{O}_1(\mathbf{Z}) = \mathbf{Z}\mathbf{H}\mathbf{H}^T - \mathbf{X}\mathbf{H}^T \tag{41}$$

$$\nabla^2 \mathcal{O}_1(\mathbf{Z}) = \mathbf{H}\mathbf{H}^T \tag{42}$$

$$\nabla \mathcal{O}_2(\mathbf{H}) = \mathbf{Z}^T\mathbf{Z}\mathbf{H} - \mathbf{Z}^T\mathbf{X} + \alpha\mathbf{H}\mathbf{L}_w - \beta\mathbf{H}\mathbf{L}_b \tag{43}$$

$$\nabla^2 \mathcal{O}_2(\mathbf{H}) = \mathbf{Z}^T\mathbf{Z} \otimes \mathbf{I}_L + \alpha\mathbf{I}_M \otimes \mathbf{L}_w - \beta\mathbf{I}_M \otimes \mathbf{L}_b \tag{44}$$

where $\otimes$ denotes the Kronecker product operation. Consequently, inequality (40) that drives the evaluation of the optimum learning step parameter α_t during the optimization of the weights matrix $\mathbf{H}$ can be rewritten as

$$\begin{aligned}(1-\sigma)\mathrm{Tr}[\nabla \mathcal{O}_2(\mathbf{H}^{(t-1)})^T(\mathbf{H}^{(t)} - \mathbf{H}^{(t-1)})] \\ + \frac{1}{2}\mathrm{vec}(\mathbf{H}^{(t)} - \mathbf{H}^{(t-1)})^T \nabla^2 \mathcal{O}_2(\mathbf{H}^{(t-1)}) \\ \times \mathrm{vec}(\mathbf{H}^{(t)} - \mathbf{H}^{(t-1)}) \leq 0\end{aligned} \tag{45}$$

where vec(.) denotes an operator that converts a matrix into a vector by stacking its columns.

E. Solving SDNMF With Nesterov's Optimal Gradient Method and Unifying With NPAF Framework

To avoid the costly line search in the Armijo rule, which requires the evaluation of the large matrices $\mathbf{I}_M \otimes \mathbf{L}_w$ and $\mathbf{I}_M \otimes \mathbf{L}_b$ in (44), we optimize SDNMF exploiting the efficient NeNMF solver in [25]. NeNMF employs the Nesterov's optimal gradient method and at each iteration obtains the approximate solution by computing the projected gradient on a search point identified by linearly combining the two latest approximate solutions. Moreover, NeNMF determines the appropriate step size by the Lipchitz constant which is

less computationally expensive than the line search. To solve SDNMF problem similarly to NeNMF, we express its cost function in (22) as follows:

$$\mathcal{O}_{SDNMF}(\mathbf{X}||\mathbf{ZH}) = \frac{1}{2}||\mathbf{X}-\mathbf{ZH}||_F^2 + \frac{\alpha}{2}\mathrm{Tr}[\mathbf{H}(\mathbf{L}_w - \frac{\beta}{\alpha}\mathbf{L}_b)\mathbf{H}^T] \tag{46}$$

The optimization of (46) with respect to $\mathbf{Z}$ is similar to the methodology presented in [25]. To optimize (46) for $\mathbf{H}$ we consider the subproblem generated by fixing variable $\mathbf{Z}$, compute its gradient which is given by (43) and determine the Lipchitz constant L as a linear combination of each Lipchitz constant of each part of (46), thus $L = ||\mathbf{Z}^T\mathbf{Z}||_F + \alpha||\mathbf{L}_w - \frac{\beta}{\alpha}\mathbf{L}_b||_F$. By replacing L and $\nabla\mathcal{O}_2(\mathbf{H})$ in Algorithm 1 in [25] we can solve SDNMF using Nesterov's optimal gradient method. We call this algorithm NeSDNMF in the rest of the manuscript.

In [12] a unified framework for various NMF-based methods called Nonnegative patch alignment framework (NPAF) has been proposed, that uses a fast gradient descent optimization algorithm. NPAF framework considers NMF-based optimization problems of the following form:

$$\mathcal{O}_{NPAF}(\mathbf{X}||\mathbf{ZH}) \triangleq \mathcal{O}_{KL}(\mathbf{X}||\mathbf{ZH}) + \frac{\alpha}{2}\mathrm{Tr}[\mathbf{HLH}^T] \tag{47}$$

where $\mathbf{L}$ is a symmetric positive semidefinite patch alignment matrix, different for each specialized algorithm. To unify SDNMF in the NPAF framework we replace in (47) the alignment matrix $\mathbf{L}$ that encodes the discriminative information by the considered in SDNMF discriminant term $\mathbf{L}_w - \frac{\beta}{\alpha}\mathbf{L}_b$. Since matrices $\mathbf{L}_w$ and $\mathbf{L}_b$ are symmetric and positive semidefinite SDNMF can be directly incorporated into the NPAF framework and optimized by the proposed in [12] generative multiplicative or fast gradient descent update rules.

F. Subclass Discriminant Kernel NMF Algorithm (SDKNMF)

In order to model nonlinearities in the extracted image features, we derive SDNMF nonlinear counterpart called SDKNMF. The problem at hand can be summarized as follows: approximate a set of nonlinear nonnegative training sample vectors mapped on a polynomial feature space, using a linear combination of appropriately weighted nonlinear nonnegative basis vectors mapped on the same polynomial feature space in a discriminant manner. Next, we shall only demonstrate the optimization of the SDKNMF problem, considering projections of the available training data to polynomial feature spaces, exploiting arbitrary degree polynomial kernel functions of the form $k(\mathbf{x}_i, \mathbf{x}_j) = \left(\mathbf{x}_i^T\mathbf{x}_j\right)^d$. However, it is straightforward to extend SDKNMF, such as to exploit different Mercer's kernels, using the methodology presented in [20].

The optimization problem for the polynomial SDKNMF algorithm is formulated as follows:

$$\min_{\mathbf{Z},\mathbf{H}} \mathcal{O}\left(\mathbf{X}^\phi||\mathbf{Z}^\phi\mathbf{H}\right) + \frac{\alpha}{2}\mathrm{Tr}[\mathbf{HL}_w\mathbf{H}^T] - \frac{\beta}{2}\mathrm{Tr}[\mathbf{HL}_b\mathbf{H}^T] \tag{48}$$
$$\text{subject to:} \quad z_{i,k} \geq 0 \quad \text{and} \quad h_{k,j} \geq 0 \quad \forall i,j,k$$

which is solved using projected gradients in order to ensure limit point stationarity. It should be noted that the previously presented methodology for the optimization of PGSDNMF algorithm is valid only for linear kernels of the form $k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T\mathbf{x}_j$ since in this case the cost function in (48) is quadratic in terms of $\mathbf{Z}$. In the general case when $d \geq 2$, the expansion performed around the current solution estimate $\mathbf{Z}^{(t-1)}$ in (35), considering up to quadratic terms, is not valid.

We similarly consider two subproblems from (48) where for each one either variable $\mathbf{Z}$ or $\mathbf{H}$ is kept fixed. The iterative process for optimizing with respect to $\mathbf{H}$ applies the update rule in (39) where the involved first and second order partial derivatives of the cost function are evaluated as

$$\nabla\mathcal{O}\left(\mathbf{X}^\phi||\mathbf{Z}^\phi\mathbf{H}\right) = \mathbf{K}_{z,z}\mathbf{H} - \mathbf{K}_{z,x} + \alpha\mathbf{HL}_w - \beta\mathbf{HL}_b \tag{49}$$
$$\nabla^2\mathcal{O}\left(\mathbf{X}^\phi||\mathbf{Z}^\phi\mathbf{H}\right) = \mathbf{K}_{z,z}\otimes\mathbf{I}_L + \alpha\mathbf{I}_M\otimes\mathbf{L}_w - \beta\mathbf{I}_M\otimes\mathbf{L}_b. \tag{50}$$

The learning step parameter α_t is similarly determined using (45) and a stationarity condition check step is performed in order to verify that the projected gradient at the reached limit point is sufficiently close to zero.

Respectfully, optimization for $\mathbf{Z}$ is performed by iterating the update rule in (33), while the optimal learning step parameter is now determined using (34) instead of (36), since the cost function for different Mercer's kernels is no longer quadratic in terms of $\mathbf{Z}$ and thus inequality (36) is not valid. Considering polynomial kernel functions of arbitrary degree the involved in (34) first order partial derivative with respect to $\mathbf{Z}$, is evaluated as

$$\nabla\mathcal{O}\left(\mathbf{X}^\phi||\mathbf{Z}^\phi\mathbf{H}\right) = \mathbf{Z}\left(\mathbf{HH}^T\odot\grave{\mathbf{K}}_{z,z}\right) - \mathbf{X}\left(\mathbf{H}\odot\grave{\mathbf{K}}_{z,x}\right)^T. \tag{51}$$

As can be observed, all involved calculations can be performed using the so-called kernel trick. Details regarding the derivation of the first order partial derivative with respect to $\mathbf{Z}$, when considering polynomial kernel functions for the nonlinear mapping are available in Appendix B.

IV. Experimental Study

We compare the proposed algorithms against various NMF-based methods, such as NMF, PGNMF [24], DNMF, PGKNMF [20], NDLA [12], and GNMF [10]. Moreover, we also include in our experimental comparison linear subspace learning methods such as CDA, LDA, PCA, LPP [38], and the marginal fisher analysis (MFA) [39], which is an appropriate LDA variant that overcomes the Gaussian distributed data samples optimality assumption. For our experiments, we consider facial expression recognition on the Cohn–Kanade (CK) [40] and the Binghamton University 3-D facial expression database (BU-3DFE) [41], face recognition on the CMU-PIE [42], and Multi-PIE [43] datasets, and object recognition on the ETH-80 [44] image set. Fig. 1 shows example images from the CK dataset, depicting the seven recognized facial expressions.

A. Preprocessing of Facial Expression Data

To form our data collections from CK and BU-3DFE datasets we only acquired a single video frame from each video sequence, depicting a subject performing a facial expression at its highest intensity level. To do so, face detection was performed using the OpenCV [45] face detector and the

Fig. 1. Sample images from the Cohn–Kanade database depicting the recognized facial expressions arranged in the following order: anger, fear, disgust, happiness, sadness, surprise, and the neutral emotional state.

Fig. 2. Mean images derived from the two more distant subclasses inside each expression class. The diverge illumination conditions during facial expressions capture in the Cohn–Kanade database are evident.

resulting facial regions of interest were manually aligned with respect to the eyes position and anisotropically scaled to a fixed size of 40×30 pixels. Finally, each gray scale facial image was scanned row-wise, so as to form a feature vector.

To measure the facial expression recognition accuracy, we randomly partitioned the available samples into five-folds and a cross validation has been performed by feeding the projected discriminant facial expression representations to a linear SVM classifier. This resulted into such a test set formation where some expressive samples of an individual were left for testing, while his rest expressive images (depicting other facial expressions) were included in the training set. This fact significantly increased the difficulty of the expression recognition problem, since identity-related issues arose.

B. Cohn–Kanade Dataset

CK is among the most popular databases for benchmarking facial expression recognition algorithms. Our data collection comprised of 407 images depicting 100 subjects, posing in seven different emotional states. As can be seen in Fig. 1, CK database images depict subjects of different ethnic groups under severe illumination variations. Consequently, the data sample vectors do not necessarily correspond to compact facial expression classes. To verify this, we have considered that each class is composed of three subclasses and computed the mean expressive image for every cluster of each class. Fig. 2 shows the mean image for each facial expression considering the two more distant clusters of each class. Clearly the illumination variations are captured during clustering.

Table I summarizes the highest performance achieved by each examined method and the respective projection subspace dimensionality. All subclass discriminant algorithms (linear and nonlinear) were found to outperform in this comparison. Moreover, the superiority of the projected gradients optimization framework is also demonstrated, since both PGNMF and PGSDNMF outperformed their multiplicative counterparts. The highest measured recognition accuracy rate is 72.9% achieved by SDKNMF algorithm, considering classes partitioning into two subclasses and a second order polynomial kernel function. Regarding the baseline algorithms PCA outperformed all linear subspace methods achieving a recognition rate of 68.8%. Moreover, MFA, which does not make any assumption on the data distribution of each class, outperformed all discrimination enhancing subspace methods. Finally, NDLA which also does not assume a Gaussian data distribution inside classes outperformed DNMF. On the other hand, GNMF although it forms similar discriminant criteria to NDLA algorithm, it is specialized for clustering problems and thus it could not provide competitive classification performance.

Fig. 3 compares the basis images produced from training on the CK database PGNMF and PGSDNMF algorithms, considering for the latter partitioning of each expression class into two subclasses. Both methods have been trained to find the optimal projection matrix to a subspace of equal dimensionality. As can be seen, the basis images extracted by PGNMF are less sparse and have a rather holistic appearance, compared to those generated by PGSDNMF. More precisely, PGSDNMF produced a few holistic basis images that highlight facial parts, common across all classes, which remain unaltered across any facial expression formation and correspond to facial areas around the nose, the forehead, and the cheeks. These bases as we have experimentally verified significantly affect the reconstruction error. The majority of PGSDNMF bases are sparse and localized around mouth, eyes, and eyebrows and highlight characteristic facial parts unique for each facial expression, such as the mouth shape at surprise expression or the raised or lowered lip corners characteristic of the happiness or sadness facial expression, respectively. These bases although do not influence the reconstruction error, however, they significantly affect the added discriminant terms since they possess valuable information for facial expression discrimination. This observation reveals that the proposed method successfully decomposed each facial image into its discriminant parts, which justifies its superior recognition performance.

C. BU-3DFE Dataset

The dataset we generated from BU-3DFE contains 700 images, depicting 100 subjects performing seven facial expressions. In the original data collection except of the neutral emotional state, each of the six performed facial expressions involves four intensity levels. In our experimental evaluation, we have included only the facial images at expressions apex.

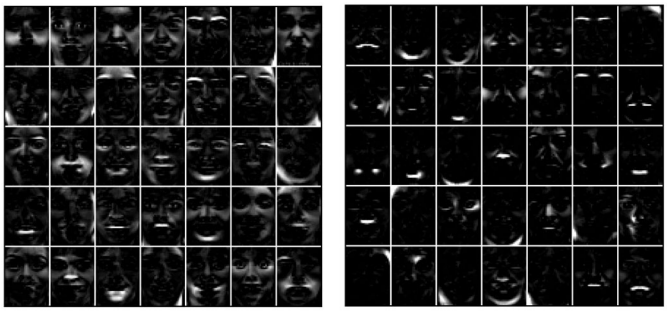
Table II presents the best average measured expression recognition accuracy rate and the respective projection subspace dimensionality, achieved by each examined method. As it can be seen the derived results are similar to those reported in the CK database. SDKNMF attained the best performance across all examined subspace methods reaching 66.4% when considering two subclasses per each expression class.

D. Face Recognition on PIE Dataset

The CMU-PIE face database contains in total 41 368 facial images depicting 68 different subjects captured under

TABLE I
BEST AVERAGE EXPRESSION RECOGNITION ACCURACY RATES (%) IN COHN–KANADE DATABASE

Linear Subspace Methods					NMF-based Methods						Proposed Methods		
LDA	PCA	CDA	LPP	MFA	NMF	PGNMF	PGKNMF	DNMF	NDLA	GNMF	SDNMF	PGSDNMF	SDKNMF
65.7	68.8	66.0	64.4	68.3	64.9	66.3	66.9	65.6	69.3	60.4	70.4	72.6	**72.9**
6	180	13	6	60	180	120	170	190	170	200	$C_r = 2$ (110)	$C_r = 2$ (190)	$\mathbf{C_r = 2(200)}$



(a) (b)

Fig. 3. Basis images derived from training in the Cohn–Kanade database algorithms (a) PGNMF and (b) PGSDNMF with C_r = 2.

variations in pose, illumination, and expression. For this experiment, we used 170 facial images for each individual captured under five near frontal poses (poses identified as C05, C07, C09, C27, and C29) under four different expressions and 43 different illumination conditions. The considered facial images were cropped, scaled to a fixed size of 32×32 pixels and gray scaled according to [10]. We randomly selected half facial images of each individual for training, while the rest were used for testing. As can be seen in Table III the proposed algorithms are more robust under mild variations in pose and expression for face recognition. PGSDNMF considering partitioning of each class into five subclasses attained the highest recognition rate, 97.7% marginally outperforming its nonlinear variant. The best recognition rates for LDA, PCA, NMF, DNMF, and SDNMF are 94.9%, 95.7%, 96.1%, 96.7%, and 97.1%, respectively.

E. Face Recognition on Multi-PIE Dataset

We also performed experiments for face recognition on the Multi-PIE dataset [43] which is collected in a setting systematically simulating the effects of pose, illumination, and expression variations and enabled us to examine the ability of the proposed methods to perform on less constrained conditions. The subset of the Multi-PIE dataset that we used in our experimental comparison contains face images from 147 subjects, captured under three different view points (i.e., $-15°$, $0°$ and $15°$), five different illumination conditions randomly selected for each subject, and six different facial expressions captured during four recording sessions. In total, we used 22 050 facial images which have been aligned and scaled to a fixed size of 40×30 pixels using the facial landmark annotations of [46]. To form our training set we randomly selected 25% of the facial images available for each subject, while the rest were used for testing. The large differences that exist between the samples of Multi-PIE span a large intra class variation and thus favor subclasses formation. Table IV summarizes the obtained results. The proposed PGSDNMF algorithm considering partitioning of each class into four subclasses attained the highest recognition rate 91.8%.

It should be noted that although PIE and Multi-PIE datasets exhibit variations in the recording settings, they both have been captured in controlled conditions and do not simulate realistically unconstrained conditions for "in-the-wild" face recognition. In realistic conditions, such as these encountered in LFW dataset [47], facial images exhibit extreme pose, illumination and background variations, occlusions, and also inaccurate alignment. In such settings NMF-based methods that exploit the image intensity domain as the underlying decomposition and classification features, are extremely sensitive since they are based on the minimization of a distance metric (e.g., Euclidean distance or the KL divergence) between the decomposed data and the derived factors. Consequently, due to the fact that all these parameters disturb these distances arbitrarily NMF-based algorithms fail to perform robustly.

F. Object Recognition on ETH-80 Dataset

ETH-80 image dataset contains 3280 images depicting 80 objects divided into eight different classes, where for each object 41 images have been captured from different view points. For this experiment, we used the cropped and scaled to a fixed size of 128×128 pixels binary images containing the contour of each object. In order to form our training set we randomly picked 25 binary images of each object, while the rest were used for testing. Since each category includes images depicting 10 different objects captured from various view angles, data samples inside classes span large in-class variations, forming various subclasses. As can be seen in Table V, PCA outperformed all linear subspace learning algorithms, while SDKNMF considering five subclasses per each object class produced the best results. The object recognition rates for LDA, PCA, CDA, DNMF, GNMF, and SDKNMF were 75.7%, 85.9%, 81.2%, 80.1%, 77.4%, and 87.1%, respectively.

To demonstrate the data clustering effect in SDNMF algorithms performance, we recorded the attained recognition rate for different parameter C_r values. As it can be seen in Fig. 4 SDNMF efficacy initially increases as we partition each class from 2 up to 5 subclasses, where our algorithm attained its best performance, while further partitioning classes results in reduced recognition accuracy. This is attributed to the fact that since training samples per subclass are limited

TABLE II
BEST AVERAGE EXPRESSION RECOGNITION ACCURACY RATES (%) IN BU-3DFE DATASET

Linear Subspace Methods					NMF-based Methods						Proposed Methods		
LDA	PCA	CDA	LPP	MFA	NMF	PGNMF	PGKNMF	DNMF	NDLA	GNMF	SDNMF	PGSDNMF	SDKNMF
54.6	64.4	59.3	55.3	58.4	58.7	59.7	62.6	63.4	61.7	56.3	64.1	64.6	**66.4**
6	100	13	6	20	150	110	200	180	180	190	$C_r = 2$ (120)	$C_r = 2$ (70)	$\mathbf{C_r = 2(190)}$

TABLE III
BEST FACE RECOGNITION ACCURACY RATES (%) IN PIE IMAGE DATABASE

Linear Subspace Methods					NMF-based Methods						Proposed Methods		
LDA	PCA	CDA	LPP	MFA	NMF	PGNMF	PGKNMF	DNMF	NDLA	GNMF	SDNMF	PGSDNMF	SDKNMF
94.9	95.7	95.1	95	93.1	96.1	96.5	93.9	96.7	93.7	94.4	97.1	**97.7**	97.5
67	300	271	67	190	200	120	300	200	160	100	$C_r = 5$ (200)	$\mathbf{C_r = 5(120)}$	$C_r = 5$(300)

TABLE IV
BEST FACE RECOGNITION ACCURACY RATES (%) IN MULTI-PIE DATABASE

Linear Subspace Methods					NMF-based Methods						Proposed Methods		
LDA	PCA	CDA	LPP	MFA	NMF	PGNMF	PGKNMF	DNMF	NDLA	GNMF	SDNMF	PGSDNMF	SDKNMF
81.5	86.7	84.6	81.8	84.0	87.3	87.4	87.0	88.2	90.4	82.5	89.6	**91.8**	90.4
146	300	300	146	300	200	120	250	300	100	250	$C_r = 4$ (300)	$\mathbf{C_r = 4(250)}$	$C_r = 4$(300)

TABLE V
BEST OBJECT RECOGNITION ACCURACY RATES (%) IN ETH-80 IMAGE DATABASE

Linear Subspace Methods					NMF-based Methods						Proposed Methods		
LDA	PCA	CDA	LPP	MFA	NMF	PGNMF	PGKNMF	DNMF	NDLA	GNMF	SDNMF	PGSDNMF	SDKNMF
75.7	85.9	81.2	75	74.6	81.3	85.2	85.4	80.1	82.6	77.4	86.7	86.7	**87.1**
7	60	31	7	240	300	60	250	300	100	250	$C_r = 5$ (300)	$C_r = 5$ (100)	$\mathbf{C_r = 5(250)}$

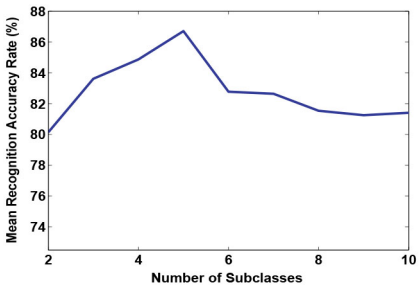


Fig. 4. Object recognition rate on ETH-80 dataset versus the number of subclasses each object category is partitioned to.

subclass covariance matrices evaluated on few examples are poorly estimated which affects the correctness of the identified projection directions [34], [48].

G. Algorithms Computational Complexity and Convergence

To investigate the ability of the proposed SDNMF and PGSDNMF algorithms to minimize the considered cost function in (23), with respect to the performed iteration rounds, we have applied both algorithms to factorize a dense data matrix composed of all expressive images in the CK dataset, considering two subclasses partitioning of each expression class and setting the projection subspace dimensionality equal to 50. Moreover, parameters α and β were set to 0.5 and 0.9, respectively, while both algorithms were initialized using the same randomly generated matrices. Fig. 5 shows the objective function value reduction per iteration, denoting the quality of the approximation, for each algorithm. As it can be observed PGSDNMF reduces the objective function in each iteration round more aggressively and converges in fewer iterations than its multiplicative counterpart.

Convergence to a stationary point of the objective function is also crucial since it determines the quality of the reached solution as well as algorithms execution time. This can be tested by checking the KKT conditions for the optimization problem in (24). The KKT conditions for the basis images matrix $\mathbf{Z}$ can be written as

$$\min\left(\mathbf{Z}, \nabla\mathcal{O}_1(\mathbf{Z})\right) = 0 \tag{52}$$

which states that both $\mathbf{Z}$ and $\nabla\mathcal{O}_1(\mathbf{Z})$ should be component-wise nonnegative and at least one of them is allowed to be zero (similarly we can form the KKT conditions for the weights matrix $\mathbf{H}$). Consequently, the KKT residual norm which can

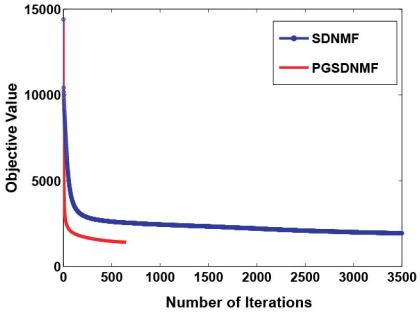


Fig. 5. Objective function value versus the number of iterations for the SDNMF and PGSDNMF algorithms.

TABLE VI
CONVERGENCE PERFORMANCE OF SDNMF, PGSDNMF, AND NESDNMF ALGORITHMS ON COHN–KANADE DATASET

	KKT Residual	# Iterations
SDNMF	39.8	10,000
PGSDNMF	3.7	100
NeSDNMF	0.8	1,000

be defined as the ℓ_l norm of (52) should tend to zero. We have investigated the convergence performance to a stationary point of the SDNMF, PGSDNMF, and NeSDNMF optimization algorithms using the data samples of CK dataset, while initialized all algorithms using the same randomly generated matrices and set to all the same parameter values. To investigate stationarity of the reached limit point we added the KKT residual norms computed for each optimized variable per iteration for each algorithm. Table VI summarizes the obtained results showing the total KKT residual and the number of iterations performed by each algorithm. As it can be seen the KKT residual norms of the PGSDNMF and the NeSDNMF algorithms are significantly smaller than that of the SDNMF. This demonstrates that the reached solution of both algorithms is closer to the stationary point of (22). Moreover, NeSDNMF using Nesterov's optimal gradient method demonstrated the best convergence performance.

To reveal the computational requirements of each method we measured the computational complexity per iteration for the derived update rules in (30) and (39) by counting the number of arithmetic operations required and summarized the results using the big O notation. Since the multiplicative updates operate on each matrix element, while the projected gradients updates perform optimization on a matrix level, in order to perform a fair comparison we measure the computational cost required by the two methods in order to update matrix $\mathbf{H}$ for a single iteration.

For both algorithms the computational cost for data clustering using the NN clustering algorithm is $O(L^2F)$. Moreover, we require LC floating point operations to compute each graph Laplacian matrix $\mathbf{L}_w$ and $\mathbf{L}_b$. Based on the update rule in (30) for each iteration the required cost for the SDNMF algorithm is $O(FLM)$. Consequently, requiring ρ iterations till convergence the overall cost is $O(\rho FLM + L^2F + LC)$. For PGSDNMF based on the alternative projective gradient approach and applying [24, Algorithm 4] to determine properly the learning rate parameter α_t the complexity is $O(FLM^2 + t \times rML^2)$ where t is the number of iterations performed for the minimization of the subproblem in (32) and r is the average number of iterations performed for finding an appropriate α_t. Consequently, the cost for a single update of the PGSDNMF algorithm is more expensive than that required by the SDNMF. However, the number of iterations required are significantly less. The overall cost till convergence for PGSDNMF is $O\left(\rho(FLM^2 + t \times rML^2) + L^2F + LC\right)$.

TABLE VII
TRAINING TIME IN SECONDS REQUIRED BY NMF, PGNMF, SDNMF, PGSDNMF, AND PCA ON COHN–KANADE DATASET

Dimensionality						
Input	Projection	NMF	PGNMF	SDNMF	PGSDNMF	PCA
1200	50	17	13	58	85	0.53

In Table VII we show the recorded in MATLAB CPU training time, measured in seconds, required by NMF, PGNMF, SDNMF, PGSDNMF, and PCA algorithms. As expected, PCA attained the shortest training time since it solves a generalized eigenvalue problem, while all NMF-based algorithms are iterative optimization methods, which are computationally expensive. Moreover, the difference in the training time between PGSDNMF and PGNMF algorithms is attributed to the involved Kronecker product operation that significantly increases the size of matrices involved in the computations of the first method.

H. Parameters Selection

The proposed update rules involve parameter C_r that affects the imposed discriminant factors and also α and β that regulate their contribution in the cost function. Although there are various methods proposed in the literature that attempt to determine the optimal data clustering setting by optimizing a specific formed criterion [49], [35], in this paper since we are interested in enhancing classes discrimination, thus increasing classification performance, we employ a similar approach to [34] and seek to determine C_r as well as parameters α and β with respect to the reached classification accuracy. Thus, we seek the C_r, α and β values for which SDNMF achieves the highest recognition rate. To do so, we performed cross validation where first we determined the optimal C_r value, while considering equal contribution of the discriminant factors setting $\alpha = \beta = 1$, and subsequently, we identified the optimal α and β values for that clustering setting. More precisely, to determine C_r, we exploited the training set in order to train our algorithms considering different values for C_r (ranging from 2 to 5 for the face databases and from 2 to 10 for the ETH-80). The range of the examined C_r values was selected such as to guarantee that the number of samples per subclass is sufficiently large (more than 10). Unfortunately, searching for all possible number of subclasses is computationally infeasible. Thus, in order to burden the computational cost we performed validation assuming that each class is composed of the same number of subclasses. Subsequently, the reached classification accuracy for each examined C_r value was measured on the

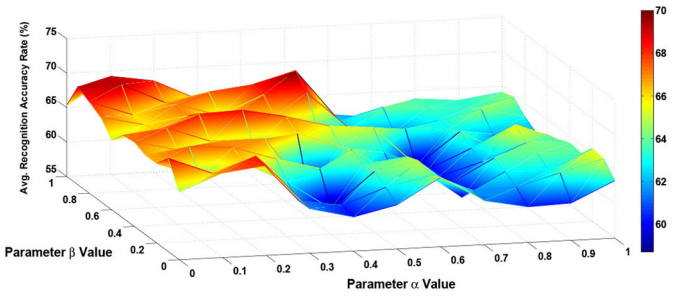


Fig. 6. SDNMF mean expression recognition rate in Cohn–Kanade database after five random starts versus the parameters α and β.

training set and the highest performing subclass partitioning setting was selected.

Parameters α and β were similarly determined through a validation stage performing a grid search, while considering the optimal clustering setting identified during the previous step. More precisely, for the facial expression recognition experiments on the CK database we trained SDNMF considering $C_r = 2$, which was identified during the previous step and set values to parameters α and β in the range $[0, 1]$. Fig. 6 shows the average reached expression recognition rates of SDNMF in CK after five random starts for each different set of parameters value. As it can be seen SDNMF performs better when α is varying within $[0, 0.5]$ and β within $[0.6, 1]$. The highest achieved recognition rate was attained for $\alpha = 0.5$ and $\beta = 0.9$ which were also the parameters value applied in experiments on both facial expression databases. A similar procedure was also applied for the ETH-80 and PIE databases, where for the first, setting in our algorithm $C_r = 5$, $\alpha = 0.4$ and $\beta = 0.6$ resulted to the best performance, while for the latter the selected parameter values were $C_r = 5$, $\alpha = 0.2$, and $\beta = 0.9$. DNMF parameters have been similarly selected using cross validation and performing a grid search in the range $[0.1, 0.5]$ according to [15]. Thus, in DNMF we set $\alpha = 0.1$ and $\beta = 0.1$ on all experiments on facial image data, while on ETH-80 dataset we set $\alpha = 0.1$ and $\beta = 0.3$. Finally, the optimal projection subspace dimensionality for the proposed algorithms as well as, for all competing algorithms in the experimental comparison has been similarly determined during validation. To do so, we have performed a grid search in the interval [50 , 300] and the subspace dimensionality, that resulted to the best recognition rate on the validation set, were subsequently adopted for the test set.

V. Conclusion

In real world applications data distribution usually does not correspond to a compact set per class, but data form various subclasses. Inspired by this observation, we investigated the use of CDA-inspired discriminant constraints in the NMF cost function, resulting in the SDNMF algorithm. SDNMF addresses the general problem of finding discriminant projections that enhance class separability by minimizing the scatter within every subclass. To solve the SDNMF minimization problem, we developed novel multiplicative update rules that consider not only sample class labels but also their subclass origin. Moreover, optimization was performed using a projected gradients framework, in order to exploit its strong optimization properties. Finally, the nonlinear counterpart of the proposed method considering projections in nonlinear polynomial feature spaces has been also investigated. We compared the performance of the proposed algorithms with that of various state-of-the-art linear subspace learning methods for facial expression, face, and object recognition verifying their effectiveness.

Appendix A
Proof of Convergence

Theorem 1: The objective function in (23) is non-increasing under the element-wise update rule in (30).

To prove Theorem 1, we define an appropriate auxiliary function G which bounds the objective function from above and also satisfies the condition $G(\mathbf{H}, \mathbf{H}) = \mathcal{O}_{SDNMF}(\mathbf{H})$. Using such an auxiliary function G we can show that the update rule

$$\mathbf{H}^{(t)} = \arg\min_{\mathbf{H}} G(\mathbf{H}, \mathbf{H}^{(t-1)}) \tag{53}$$

will never increase the objective function, since the following inequality holds:

$$\begin{aligned}\mathcal{O}_{SDNMF}\left(\mathbf{H}^{(t)}\right) &\leq G\left(\mathbf{H}^{(t)}, \mathbf{H}^{(t-1)}\right) \\ &\leq G\left(\mathbf{H}^{(t-1)}, \mathbf{H}^{(t-1)}\right) = \mathcal{O}_{SDNMF}\left(\mathbf{H}^{(t-1)}\right).\end{aligned} \tag{54}$$

Lemma: $G(h, h_{k,j}^{(t-1)})$ is an auxiliary function for $F_{h_{k,j}}$, which is the part of (23) that is only relevant to $h_{k,j}$

$$\begin{aligned}G(h, h_{k,j}^{(t-1)}) &= F_{h_{k,j}}(h_{k,j}^{(t-1)}) + F'_{h_{k,j}}(h_{k,j}^{(t-1)})(h - h_{k,j}^{(t-1)}) \\ &+ \frac{[\mathbf{Z}^T\mathbf{Z}\mathbf{H}]_{k,j} + \alpha[\mathbf{H}\mathbf{L}_w]_{k,j} + \beta[\mathbf{H}\mathbf{E}(\mathbf{1} - \sum_{r=1}^{C_r} \mathbf{e}_r^T\mathbf{e}_r)\mathbf{E}]_{k,j}}{2h_{k,j}^{(t-1)}}(h - h_{k,j}^{(t-1)})^2\end{aligned} \tag{55}$$

where $\mathbf{E} \in \Re^{L \times L}$ is a diagonal matrix containing vector $\mathbf{e}$ on its main diagonal defined as $\mathbf{E} = \text{diag}(\mathbf{e})$.

Proof: Let us denote with $F'_{h_{k,j}}$ and $F''_{h_{k,j}}$ the first and second order derivatives of $F_{h_{k,j}}$ with respect to $h_{k,j}$ evaluated as

$$F'_{h_{k,j}} = [\mathbf{Z}^T\mathbf{Z}\mathbf{H}]_{k,j} - [\mathbf{Z}^T\mathbf{X}]_{k,j} + \alpha[\mathbf{H}\mathbf{L}_w]_{k,j} - \beta[\mathbf{H}\mathbf{L}_b]_{k,j} \tag{56}$$

$$F''_{h_{k,j}} = [\mathbf{Z}^T\mathbf{Z}]_{k,k} + \alpha[\mathbf{L}_w]_{j,j} - \beta[\mathbf{L}_\mathbf{b}]_{j,j} \tag{57}$$

Obviously, according to the definition of the auxiliary function in (55) it holds $G(h, h) = F_{h_{k,j}}(h)$. Consequently, we only need to show that $G(h, h_{k,j}^{(t-1)}) \geq F_{h_{k,j}}(h)$. To do so, we compare $G(h, h_{k,j}^{(t-1)})$ with the up to second order Taylor series expansion of $F_{h_{k,j}}(h)$ defined as

$$\begin{aligned}F_{h_{k,j}}(h) &= F_{h_{k,j}}(h_{k,j}^{(t-1)}) + F'_{h_{k,j}}(h - h_{k,j}^{(t-1)}) \\ &+ \frac{1}{2}F''_{h_{k,j}}(h - h_{k,j}^{(t-1)})^2.\end{aligned} \tag{58}$$

Substituting (57) into (58) and comparing it with (55), we derive that instead of showing that $G(h, h_{k,j}^{(t-1)}) \geq F_{h_{k,j}}(h)$ we

can equivalently prove that

$$\frac{[\mathbf{Z}^T\mathbf{ZH}]_{k,j}+\alpha[\mathbf{HL}_w]_{k,j}+\beta[\mathbf{HE}(\mathbf{1}-\sum_{r=1}^{C_r}\mathbf{e}_r^T\mathbf{e}_r)\mathbf{E}]_{k,j}}{h_{k,j}^{(t-1)}} \geq F'_{h_{k,j}} . \quad (59)$$

To prove inequality (59) we compare separately each term

$$[\mathbf{Z}^T\mathbf{ZH}]_{k,j} = \sum_{l=1}^{L}[\mathbf{Z}^T\mathbf{Z}]_{k,k}h_{k,l}^{(t-1)} \geq [\mathbf{Z}^T\mathbf{Z}]_{k,k}h_{k,j}^{(t-1)} \quad (60)$$

$$\alpha[\mathbf{HL}_w]_{k,j} = \alpha\sum_{l=1}^{M}h_{k,l}^{(t-1)}[\mathbf{L}_w]_{l,j} \geq \alpha h_{k,j}^{(t-1)}[\mathbf{L}_w]_{j,j}. \quad (61)$$

To complete the proof we need to show that

$$\begin{aligned}&\left[\mathbf{HE}(\mathbf{1}-\sum_{r=1}^{C_r}\mathbf{e}_r^T\mathbf{e}_r)\mathbf{E}\right]_{k,j} \geq -h_{k,j}^{(t-1)}[\mathbf{L}_b]_{j,j} \Leftrightarrow \\ &\left[\mathbf{HE}(\mathbf{1}-\sum_{r=1}^{C_r}\mathbf{e}_r^T\mathbf{e}_r)\mathbf{E}\right]_{k,j} \geq -h_{k,j}^{(t-1)}\Big[\sum_{r=1}^{n}\sum_{\theta=1}^{C_r}\frac{C-C_r}{N_{r,\theta}^2}\mathbf{e}_{r,\theta}^T\mathbf{e}_{r,\theta}\Big]_{j,j}\end{aligned} \quad (62)$$

since $\left[\mathbf{E}(\mathbf{1}-\sum_{r=1}^{C_r}\mathbf{e}_r^T\mathbf{e}_r)\mathbf{E}\right]_{j,j} = 0$ given that matrix $\left[\sum_{r=1}^{C_r}\mathbf{e}_r^T\mathbf{e}_r\right]_{j,j}$ is block diagonal with all its diagonal elements equal to one. Consequently, inequality (62) is simplified to

$$\begin{aligned}\sum_{l=1}^{M}h_{k,l}^{(t-1)}&\left[\mathbf{E}(\mathbf{1}-\sum_{r=1}^{C_r}\mathbf{e}_r^T\mathbf{e}_r)\mathbf{E}\right]_{l,j} \\ &+h_{k,j}^{(t-1)}\Big[\sum_{r=1}^{n}\sum_{\theta=1}^{C_r}\frac{C-C_r}{N_{r,\theta}^2}\mathbf{e}_{r,\theta}^T\mathbf{e}_{r,\theta}\Big]_{j,j} \geq 0\end{aligned} \quad (63)$$

which is valid since $\left[\sum_{r=1}^{n}\sum_{\theta=1}^{C_r}\frac{C-C_r}{N_{r,\theta}^2}\mathbf{e}_{r,\theta}^T\mathbf{e}_{r,\theta}\right]_{j,j} \geq 0$, since $C \geq C_r$ and also $\left[\mathbf{E}(\mathbf{1}-\sum_{r=1}^{C_r}\mathbf{e}_r^T\mathbf{e}_r)\mathbf{E}\right]_{l,j} \geq 0$. Summing up all the above inequalities completes the proof. ■

Proof of Theorem 1: Consequently, (55) is an auxiliary function of (23) and $\mathcal{O}_{SDNMF}$ is non-increasing under the update in (30).

Appendix B
First Order Partial Derivatives With Respect to Z Considering Arbitrary Degree Polynomial Kernels

The first order partial derivative of $\mathcal{O}\left(\mathbf{X}^{\phi}||\mathbf{Z}^{\phi}\mathbf{H}\right)$ with respect to $z_{k,l}$ considering $\mathbf{H}$ fixed, is evaluated as follows:

$$\begin{aligned}\frac{\partial\mathcal{O}\left(\mathbf{X}^{\phi}||\mathbf{Z}^{\phi}\mathbf{H}\right)}{\partial z_{k,l}} = \sum_{i=1}^{L}\Bigg(&-h_{l,i}\frac{\partial k(\mathbf{z}_l,\mathbf{x}_i)}{\partial z_{k,l}} \\ &+\left(\sum_{j=1}^{M}h_{l,i}h_{j,i}\times\frac{\partial k(\mathbf{z}_i,\mathbf{z}_l)}{\partial z_{k,l}}\right. \\ &\left.+\sum_{j\neq l}^{M}h_{l,i}h_{j,i}\frac{\partial k(\mathbf{z}_j,\mathbf{z}_l)}{\partial z_{k,l}}\right)\Bigg).\end{aligned} \quad (64)$$

Considering a polynomial kernel its partial derivative with respect to $z_{k,l}$ is

$$\frac{\partial k(\mathbf{z}_j,\mathbf{z}_l)}{\partial z_{k,l}} = \frac{\partial\left(\sum_{i=1}^{F}z_{i,j}z_{i,l}\right)^d}{\partial z_{k,l}} = dz_{k,j}\left(\mathbf{z}_j^T\mathbf{z}_l\right)^{d-1} . \quad (65)$$

Consequently, replacing (65) into (64) we derive $\nabla\mathcal{O}\left(\mathbf{X}^{\phi}||\mathbf{Z}^{\phi}\mathbf{H}\right)$ as

$$\begin{aligned}\frac{\partial\mathcal{O}\left(\mathbf{X}^{\phi}||\mathbf{Z}^{\phi}\mathbf{H}\right)}{\partial z_{k,l}} = &-\sum_{i=1}^{L}h_{l,i}x_{k,i}d\left(\mathbf{x}_i^T\mathbf{z}_l\right)^{d-1} \\ &+\sum_{i=1}^{L}\sum_{j=1}^{M}h_{l,i}h_{j,i}z_{k,j}d\left(\mathbf{z}_j^T\mathbf{z}_l\right)^{d-1}\end{aligned} \quad (66)$$

which in matrix form can be written as

$$\nabla\mathcal{O}\left(\mathbf{X}^{\phi}||\mathbf{Z}^{\phi}\mathbf{H}\right) = \mathbf{Z}\left(\mathbf{HH}^T\odot\grave{\mathbf{K}}_{z,z}\right)-\mathbf{X}\left(\mathbf{H}\odot\grave{\mathbf{K}}_{z,x}\right)^T . \quad (67)$$

References

[1] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, May 1999.

[2] I. Jolliffe, *Principal Component Analysis*. New York, NY, USA: Springer-Verlag, 1986.

[3] A. Bell and T. Sejnowski, "The 'independent components' of natural scenes are edge filters," *Vis. Res.*, vol. 37, no. 23, pp. 3327–3338, 1997.

[4] P. Comon, "Independent component analysis, a new concept?" *Signal Process.*, vol. 36, no. 3, pp. 287–314, 1994.

[5] G. Golub and C. Van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD, USA: Johns Hopkins, 1996.

[6] P. Hoyer, "Non-negative matrix factorization with sparseness constraints," *J. Mach. Learn. Res.*, vol. 5, pp. 1457–1469, Jan. 2004.

[7] S. Li, X. Hou, H. Zhang, and Q. Cheng, "Learning spatially localized, parts-based representation," in *Proc. IEEE CVPR*, 2001, pp. 207–212.

[8] Z. Yang and E. Oja, "Linear and nonlinear projective nonnegative matrix factorization," *IEEE Trans. Neural Netw.*, vol. 21, no. 5, pp. 734–749, May 2010.

[9] C. Ding, T. Li, and M. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 45–55, Jan. 2010.

[10] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized non-negative matrix factorization for data representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, Aug. 2011.

[11] N. Guan, D. Tao, Z. Luo, and B. Yuan, "Manifold regularized discriminative nonnegative matrix factorization with fast gradient descent," *IEEE Trans. Image Process.*, vol. 20, no. 7, pp. 2030–2048, Jul. 2011.

[12] N. Guan, D. Tao, Z. Luo, and B. Yuan, "Non-negative patch alignment framework," *IEEE Trans. Neural Netw.*, vol. 22, no. 8, pp. 1218–1230, Aug. 2011.

[13] T. Zhang, B. Fang, Y. Tang, G. He, and J. Wen, "Topology preserving non-negative matrix factorization for face recognition," *IEEE Trans. Image Process.*, vol. 17, no. 4, pp. 574–584, Apr. 2008.

[14] Y. Wang, Y. Jia, C. Hu, and M. Turk, "Non-negative matrix factorization framework for face recognition," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 19, no. 4, pp. 495–511, 2005.

[15] S. Zafeiriou, A. Tefas, I. Buciu, and I. Pitas, "Exploiting discriminant information in nonnegative matrix factorization with application to frontal face verification," *IEEE Trans. Neural Netw.*, vol. 17, no. 3, pp. 683–695, May 2006.

[16] S. Zafeiriou, "Discriminant nonnegative tensor factorization algorithms," *IEEE Trans. Neural Netw.*, vol. 20, no. 2, pp. 217–235, Feb. 2009.

[17] S. Nikitidis, A. Tefas, N. Nikolaidis, and I. Pitas, "Facial expression recognition using clustering discriminant non-negative matrix factorization," in *Proc. 18th IEEE ICIP*, Brussels, Belgium, 2011, pp. 3001–3004.

[18] I. Kotsia, S. Zafeiriou, and I. Pitas, "A novel discriminant non-negative matrix factorization algorithm with applications to facial image characterization problems," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 3, pp. 588–595, Sep. 2007.

[19] I. Buciu, N. Nikolaidis, and I. Pitas, "Nonnegative matrix factorization in polynomial feature space," *IEEE Trans. Neural Netw.*, vol. 19, no. 6, pp. 1090–1100, Jun. 2008.
[20] S. Zafeiriou and M. Petrou, "Nonlinear non-negative component analysis algorithms," *IEEE Trans. Image Process.*, vol. 19, no. 4, pp. 1050–1066, Apr. 2010.
[21] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. Boston, MA, USA: Academic Press, 1990.
[22] X. Chen and T. Huang, "Facial expression recognition: A clustering-based approach," *Pattern Recognit. Lett.*, vol. 24, nos. 9–10, pp. 1295–1302, 2003.
[23] C.-J. Lin, "On the convergence of multiplicative update algorithms for nonnegative matrix factorization," *IEEE Trans. Neural Netw.*, vol. 18, no. 6, pp. 1589–1596, Nov. 2007.
[24] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural Comput.*, vol. 19, no. 10, pp. 2756–2779, 2007.
[25] N. Guan, D. Tao, Z. Luo, and B. Yuan, "NeNMF: An optimal gradient method for non-negative matrix factorization," *IEEE Trans. Signal Process.*, vol. 60, no. 6, pp. 2882–2898, Jun. 2012.
[26] S. Nikitidis, A. Tefas, and I. Pitas, "Using subclasses in discriminant non-negative subspace learning for facial expression recognition," in *Proc. 19th EUSIPCO*, Barcelona, Spain, Aug.–Sep. 2011, pp. 1964–1968.
[27] S. Nikitidis, A. Tefas, N. Nikolaidis, and I. Pitas, "Subclass discriminant nonnegative matrix factorization for facial image analysis," *Pattern Recognit.*, vol. 45, no. 12, pp. 4080–4091, 2012.
[28] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. NIPS*, 2000, pp. 556–562.
[29] R. Fletcher, *Practical Methods of Optimization*, 2nd ed. New York, NY, USA: Wiley-Interscience, 1987.
[30] S. Nikitidis, N. Nikolaidis, and I. Pitas, "Multiplicative update rules for incremental training of multiclass support vector machines," *Pattern Recognit.*, vol. 45, no. 5, pp. 1838–1852, 2012.
[31] Z. Yang and E. Oja, "Unified development of multiplicative algorithms for linear and quadratic nonnegative matrix factorization," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 1878–1891, Dec. 2011.
[32] Y. Xue, C. Tong, Y. Chen, and W. Chen, "Clustering-based initialization for non-negative matrix factorization," *Appl. Math. Comput.*, vol. 205, no. 2, pp. 525–536, 2008.
[33] C. Boutsidis and E. Gallopoulos, "SVD based initialization: A head start for nonnegative matrix factorization," *Pattern Recognit.*, vol. 41, no. 4, pp. 1350–1362, 2008.
[34] M. Zhu and A. Martinez, "Subclass discriminant analysis," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 28, no. 8, pp. 1274–1286, Aug. 2006.
[35] A. Azran and Z. Ghahramani, "Spectral methods for automatic multiscale data clustering," in *Proc. IEEE CVPR*, 2006, pp. 190–197.
[36] C. Lin and J. Moré, "Newton's method for large bound-constrained optimization problems," *SIAM J. Optim.*, vol. 9, no. 4, pp. 1100–1127, 1999.
[37] D. Bertsekas, "On the Goldstein–Levitin–Polyak gradient projection method," *IEEE Trans. Autom. Control*, vol. 21, no. 2, pp. 174–184, Apr. 1976.
[38] X. He and P. Niyogi, "Locality preserving projections," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2003.
[39] S. Yan *et al.*, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jan. 2007.
[40] T. Kanade, J. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," in *Proc. IEEE Int. Conf. AFGR*, Grenoble, France, Mar. 2000, pp. 46–53.
[41] L. Yin, X. Wei, Y. Sun, J. Wang, and M. Rosato, "A 3D facial expression database for facial behavior research," in *Proc. IEEE Int. Conf. AFGR*, Southampton, England, Apr. 2006, pp. 211–216.
[42] 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.
[43] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-pie," *Image Vis. Comput.*, vol. 28, no. 5, pp. 807–813, 2010.
[44] B. Leibe and B. Schiele, "Analyzing appearance and contour based methods for object categorization," in *Proc. IEEE CVPR*, Jun. 2003.
[45] G. Bradski, A. Kaehler, and V. Pisarevsky, "Learning-based computer vision with intel's open source computer vision library," *Intel Tech. J.*, vol. 9, no. 2, pp. 119–130, 2005.
[46] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "A semi-automatic methodology for facial landmark annotation," in *Proc. IEEE CVPRW*. Portland, OR, USA, 2013, pp. 896–903.
[47] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Univ. Massachusetts, Amherst, MA, USA, Tech. Rep. 07-49, Oct. 2007.
[48] 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 Tech.*, vol. 14, no. 2, pp. 214–223, Feb. 2004.
[49] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern Recognit.*, vol. 36, no. 2, pp. 451–461, 2003.

Symeon Nikitidis received the B.Sc. degree in informatics from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2004, the M.Sc. degree in advanced computing from the University of Glasgow, Glasgow, U.K., in 2005, and the Ph.D. degree in informatics from the Aristotle University of Thessaloniki in 2013.

From 2006 to 2012, he was a Researcher and Teaching Assistant at the Department of Informatics, Aristotle University of Thessaloniki. Since 2012, he has been a Research Associate with the Department of Computing in Imperial College London, London, U.K. His current research interests include statistical machine learning, digital signal and image processing, pattern recognition, and computer vision.

Anastasios Tefas (M'04) received the B.Sc. and Ph.D. degrees in informatics from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 1997 and 2002, respectively.

He is currently an Assistant Professor with Department of Informatics, University of Thessaloniki, where he was a Researcher and Teaching Assistant from 1997 to 2002, a Temporary Lecturer from 2003 to 2004, and a Lecturer from 2008 to 2012. From 2006 to 2008, he was an Assistant Professor at the Department of Information Management, Technological Institute of Kavala, Kavala, Greece. His current research interests include computational intelligence, pattern recognition, statistical machine learning, digital signal and image processing, and computer vision. He has participated in 12 research projects financed by national and European funds. He has co-authored 40 journal papers, 120 papers in international conferences, and contributed seven chapters to edited books in his area of expertise. Over 2150 citations have been recorded to his publications and his H-index is 23 according to Google Scholar.

Ioannis Pitas (SM'94–F'07) received the Diploma and Ph.D. degrees in electrical engineering from the Aristotle University of Thessaloniki, Thessaloniki, Greece.

Since 1994, he has been a Professor at the Department of Informatics, Aristotle University of Thessaloniki. He has also served as a Visiting Professor at several universities. His current research interests include image/video processing, intelligent digital media, machine learning, human-centered interfaces, affective computing, computer vision, 3-D imaging, and biomedical imaging. He has published over 750 papers, contributed to 39 books in his areas of expertise, and edited or (co-)authored another nine books. He participated in 68 research and development projects, primarily funded by the European Union and was also a Principal Investigator/Researcher in 40 such projects. He has over 18 400 Google Scholar and 6250 Scopus citations to his work, and an H-index of over 64 with Google scholar and 38 with Scopus.

Prof. Pitas has also been an invited speaker and/or member of the program committee of many scientific conferences and workshops. He was an Associate Editor or Co-Editor of eight international journals and General or Technical Chair of four international conferences (including ICIP2001).