Projected Gradients for Subclass Discriminant Non-negative Subspace Learning

Symeon Nikitidis, Anastasios Tefas and Ioannis Pitas

***Abstract*—Current discriminant Non-negative Matrix Factorization methods either do not guarantee convergence to a stationary limit point or assume a compact data distribution inside classes, thus ignoring intra class variances in extracting discriminant data samples representation. To address both limitations, we regard that data inside each class has a multimodal distribution, forming various clusters 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.**

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 to seek for efficient dimensionality reduction methods for appropriate image feature extraction, which will alleviate computational complexity and boost the performance of succeeding processing algorithms. Such a popular category of methods, is the subspace image representation algorithms which aim to discover the latent image features by projecting linearly or non-linearly an image to a low dimensional subspace, where a certain criterion is optimized.

Non-negative Matrix Factorization (NMF) [1], is a popular subspace learning 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 non-negative elements. The non-negativity constraint imposed by NMF on both the latent variables and the observations is meaningful, when we operate on image data exploiting their intensities, as the underlying features are naturally non-negative. Moreover, the semantic interpretability of the non-negative subspace learning is enhanced, since this conforms nicely to identifying appropriate basic elements, corresponding to the basis images, which are added to reconstruct the original data. This non-negativity limitation 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 non overlapping. However, because the sparseness achieved by the original NMF is somewhat of a side-effect rather than a goal, caused by the imposed non-negativity constraints, different studies have attempted to control the degree to which the derived representation is sparse. Towards this direction, Hoyer in [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 original NMF. An extension of NMF that is applicable on mixed sign data has been attempted in [9], where the non-negativity constraint on the basis images has been relaxed, while the weights matrix remained positively constrained. Towards improving clustering performance, Cai et al. [10], [11] recently 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 [12], [13]. Another notable variant of NMF which retains the manifold structure of facial space, is the Topology Preserving NMF (TPNMF) proposed by Zhang et al. that is specialized for face representation and recognition [14].

Focusing on applications operating on facial image data, numerous specialized NMF decomposition variants have been proposed for face recognition [7], [14], [15], face verification [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 from the original high dimensional image space into a lower dimensional subspace where the criterion under consideration is optimized. In order to model properly the non-linearities 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 [16]. DNMF incorporates a discriminant factor 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 sample vectors of each class are generated from underlying multivariate normal distributions of 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 non-linearly separable classes cannot be treated efficiently. Unfortunately, in real world applications, data distribution usually does not correspond to compact sets. This is common e.g. in facial expression recognition, since there is no unique way that people form certain expressions and moreover, there are other factors, such as pose, texture and illumination variations that lead to expression subclasses [22]. If this fact is not properly addressed, the performance of NMF-based methods is significantly degraded [23].

To overcome the aforementioned limitations we relax the assumption that each class consists of a single compact data cluster and regard that they form 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) introduced in [22] aiming at better subclasses separation. Moreover, we extend NMF reformulating the cost function that drives the optimization process 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 available training data. To solve the SDNMF problem, we develop update rules that consider not only samples class origin but also subclasses formation inside each class. In addition, in order to exploit the well established optimization properties of [24], [25] that ensure stationarity of the reached limit point, we solve SDNMF problem using an iterative projected gradients optimization framework. Finally, we derive the non-linear counterpart of SDNMF that projects training data to high dimensionality 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 the following:

- Subclass discriminant constraints that assume multimodal data distribution are incorporated in the NMF cost function, resulting in a specialized NMF based method called Subclass Discriminant NMF.
- To solve SDNMF, novel update rules under two different optimization frameworks are proposed and their optimization properties and proof of convergence are exhibited.
- The non-linear counterpart of SDNMF algorithm that considers projections in high dimensional Hilbert spaces is demonstrated.
- A thorough experimental study on various image recognition problems is performed, comparing the proposed methods with current state-of-the-art linear and non-linear dimensionality reduction algorithms.

The rest of the paper is organized as follows. The linear and non-linear NMF 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 non-linear 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], [27].

II. Linear and Non-linear NMF and Its Discriminant Variant

In this section, we briefly present the linear and non-linear NMF decomposition concept and also review DNMF algorithm. In the following, without loosing generality, we shall assume that the decomposed data are images, although, the techniques that will be described can be applied to any kind of non-negative 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 $\mathbf{Z} \in R_+^{F\times M}$ 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 $\mathbf{H}$. Useful factorizations appear when the linear transformation projects data from the original high dimensional space to a reduced dimensional subspace (i.e. $M \ll F$).

To measure the cost of the decomposition in (1), one popular approach is to use the matrix Frobenius norm square. Thus the NMF cost function $\mathcal{O}_F(\mathbf{X}||\mathbf{ZH})$ can be measured as the sum of the squared Euclidean distances between all original images in the database and their respective reconstructed versions:

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

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

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

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 to the desired factors, while guarantee 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. Non-linear NMF

The problem of Non-linear NMF (NNMF) can be summarized as follows: find a set of non-negative weights and non-negative, non-linear basis vectors such that the non-negative non-linearly mapped training data can be approximated as a linear combination of the learned non-negative non-linearly mapped basis vectors. This can be formulated as follows. Let $\phi(\mathbf{x}_i) : R_+^F \rightarrow \mathcal{H}$ be a non-linear 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:

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

where the kernel matrices are defined as:

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

Thus, NNMF solves the following optimization problem:

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

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):

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

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

where $\mathbf{\Omega}$ is a diagonal matrix, with $[\mathbf{\Omega}]_{j,j} = \sum_{k=1}^{M} h_{k,j}$ 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 [16] algorithm is an attempt to introduce discriminant constraints in the NMF decomposition cost function. To derive these the well known Fisher discriminant criterion has been exploited, which attempts to find a transformation matrix $\mathbf{\Psi}$ that maximizes the ratio defined by the traces of the between and within class scatter matrices $\acute{\mathbf{S}}_b = \mathbf{\Psi}^T\mathbf{S}_b\mathbf{\Psi}$ and $\acute{\mathbf{S}}_w = \mathbf{\Psi}^T\mathbf{S}_w\mathbf{\Psi}$ 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 algorithm minimizes the following cost function:

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

where Tr[.] is the matrix trace operator and α, β are positive constants.

III. Subclass Discriminant Non-negative 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 that solve SDNMF and also its non-linear counterpart.

A. Subclass based Discriminant Analysis

Similar to LDA, CDA seeks to determine a transformation matrix $\mathbf{\Psi}$ 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:

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

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

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

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:

$$\begin{aligned} \boldsymbol{\Sigma}_w &\triangleq \sum_{r=1}^{n} \sum_{\theta=1}^{C_r} \sum_{j=1}^{N_{r,\theta}} \left(\mathbf{h}_j - \boldsymbol{\mu}^{r,\theta}\right)\left(\mathbf{h}_j - \boldsymbol{\mu}^{r,\theta}\right)^T \\ &= \mathbf{H}\mathbf{L}_w\mathbf{H}^T \end{aligned} \quad (15)$$

and

$$\begin{aligned} \boldsymbol{\Sigma}_b &\triangleq \sum_{i=1}^{n} \sum_{r,r\neq i}^{n} \sum_{j=1}^{C_i} \sum_{\theta=1}^{C_r} \left(\boldsymbol{\mu}^{i,j} - \boldsymbol{\mu}^{r,\theta}\right)\left(\boldsymbol{\mu}^{i,j} - \boldsymbol{\mu}^{r,\theta}\right)^T \\ &= \mathbf{H}\mathbf{L}_b\mathbf{H}^T, \end{aligned} \quad (16)$$

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

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

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

Here $\text{diag}(\mathbf{e})$ denotes a function that converts vector $\mathbf{e}$ into a diagonal matrix containing its elements on the main diagonal, $\mathbf{I}_L$ 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:

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

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

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

The trace of the within subclass scatter matrix $\boldsymbol{\Sigma}_w$ can be used as an appropriate indicator of the samples dispersion inside subclasses. Minimizing its trace increases concentration of samples around their subclass mean. Similarly, $\text{Tr}[\boldsymbol{\Sigma}_b]$ indicates the dispersion of the mean vectors between all subclasses that belong to different classes. Thus, maximizing $\text{Tr}[\boldsymbol{\Sigma}_b]$ increases the difference between the means of every subclass of a certain class to every subclass of each other class.

B. SDNMF Objective Function and its Multiplicative Update Rules

Since we desire in the projection subspace to simultaneously minimize $\text{Tr}[\boldsymbol{\Sigma}_w]$ and maximize $\text{Tr}[\boldsymbol{\Sigma}_b]$, the cost function of the SDNMF algorithm is formulated as follows:

$$\begin{aligned} \mathcal{O}_{SDNMF}(\mathbf{X}||\mathbf{ZH}) &\triangleq \frac{1}{2}||\mathbf{X} - \mathbf{ZH}||_F^2 + \frac{\alpha}{2}\text{Tr}[\mathbf{H}\mathbf{L}_w\mathbf{H}^T] \\ &- \frac{\beta}{2}\text{Tr}[\mathbf{H}\mathbf{L}_b\mathbf{H}^T], \end{aligned} \quad (22)$$

where α and β are positive constants, while $\frac{1}{2}$ is used to simplify subsequent mathematical derivations. Alternatively, the SDNMF cost function can be written using matrices trace form as follows:

$$\begin{aligned} \mathcal{O}_{SDNMF}(\mathbf{X}||\mathbf{ZH}) &= \frac{1}{2}\text{Tr}[\mathbf{X}\mathbf{X}^T] - \text{Tr}[\mathbf{Z}\mathbf{H}\mathbf{X}^T] \quad (23) \\ +\frac{1}{2}\text{Tr}[\mathbf{Z}\mathbf{H}\mathbf{H}^T\mathbf{Z}^T] &+ \frac{\alpha}{2}\text{Tr}[\mathbf{H}\mathbf{L}_w\mathbf{H}^T] - \frac{\beta}{2}\text{Tr}[\mathbf{H}\mathbf{L}_b\mathbf{H}^T], \end{aligned}$$

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

Consequently, the minimization problem of SDNMF is formulated as:

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

which requires the minimization of (23) subject to the non-negativity constraints applied on the elements of both factors $\mathbf{H}$ and $\mathbf{Z}$.

In order to solve the constrained optimization problem in (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 non-negativity constraints $z_{i,k} \geq 0$, $h_{k,j} \geq 0$, respectively. Consequently, we formulate the Lagrangian function $\mathcal{L}$ as follows:

$$\begin{aligned} \mathcal{L} &= \frac{1}{2}\text{Tr}[\mathbf{X}\mathbf{X}^T] - \text{Tr}[\mathbf{Z}\mathbf{H}\mathbf{X}^T] + \frac{1}{2}\text{Tr}[\mathbf{Z}\mathbf{H}\mathbf{H}^T\mathbf{Z}^T] + \text{Tr}[\boldsymbol{\psi}\mathbf{H}^T] \\ &+ \frac{\alpha}{2}\text{Tr}[\mathbf{H}\mathbf{L}_w\mathbf{H}^T] - \frac{\beta}{2}\text{Tr}[\mathbf{H}\mathbf{L}_b\mathbf{H}^T] + \text{Tr}[\boldsymbol{\phi}\mathbf{Z}^T]. \end{aligned} \quad (25)$$

The optimization problem in equation (24) is equivalent to the minimization of the Lagrangian function $\arg\min_{\mathbf{Z},\mathbf{H}} \mathcal{L}$. By setting the partial derivatives of $\mathcal{L}$ with respect to $z_{i,k}$ and $h_{k,j}$ equal to zero and exploiting the KKT conditions, [29] we obtain the following equalities::

$$\begin{aligned} \left(\frac{\partial \mathcal{L}}{\partial h_{k,j}}\right) h_{k,j} &= [\mathbf{Z}^T\mathbf{Z}\mathbf{H}]_{k,j} h_{k,j} - [\mathbf{Z}^T\mathbf{X}]_{k,j} h_{k,j} \\ &+ \alpha[\mathbf{H}\mathbf{L}_w]_{k,j} h_{k,j} - \beta[\mathbf{H}\mathbf{L}_b]_{k,j} h_{k,j} = 0 \quad (26) \\ \left(\frac{\partial \mathcal{L}}{\partial z_{i,k}}\right) z_{i,k} &= [\mathbf{Z}\mathbf{H}\mathbf{H}^T]_{i,k} z_{i,k} - [\mathbf{X}\mathbf{H}^T]_{i,k} z_{i,k} = 0. \quad (27) \end{aligned}$$

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 formulae in (5). This can be also verified by solving (27) 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 (26) for $h_{k,j}$ we derive the proposed multiplicative update rule shown in (28). A detailed proof regarding the non-increasing behavior of (23) under the proposed update rules in (28) for $\mathbf{H}$ can be found in the Appendix A.

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 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 work. Other computationally more complex approaches to initialize the decomposition factors are based on K-means clustering [30] or SVD decomposition [31].

The optimization process successively updates variable $\mathbf{Z}$ or $\mathbf{H}$ until a stopping criterion is invoked. In this work we terminate the optimization process when the improvement in the cost function value between two successive iterations is less than 10^{-3}. Other similar stopping criteria based on monitoring the objective function improvement have been proposed in the literature [32]. Finally, in order to compute the projection to the lower dimensional feature space for an unknown test sample $\mathbf{x}_j$ and extract its discriminant representation 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 the optimization of SDNMF.

Input: Non-negative 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 $\mathbf{x}_i$ $i = 1, \ldots, L$.
Output: The basis images matrix $\mathbf{Z} \in R_+^{F\times M}$ and the weights matrix $\mathbf{H} \in R_+^{M\times L}$.
Initialize: $\mathbf{Z}^{(0)}$, $\mathbf{H}^{(0)}$ and $t = 1$.
repeat
  Update $\mathbf{H}^{(t)}$ given $\mathbf{Z}^{(t-1)}$ using (28).
  Update $\mathbf{Z}^{(t)}$ given $\mathbf{H}^{(t)}$ using (5).
  $t = t + 1$.
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 [33], [34]. In our implementation, we have considered the Nearest-Neighbor (NN) based clustering algorithm presented in [33] which is a good compromise between computation speed and clustering accuracy. Moreover, as it has been shown in [33] 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 sample vectors 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., the two sample vectors that maximize the Euclidean distance $\mathrm{argmax}_{\mathbf{x}_i,\mathbf{x}_j} ||\mathbf{x}_i - \mathbf{x}_j||_2$. 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 subsets, thus obtaining C_r subclasses.

D. Projected Gradients Subclass Discriminant Non-negative Matrix Factorization (PGSDNMF)

The derived multiplicative update rules for the evaluation of the optimal factors $\mathbf{H}$ and $\mathbf{Z}$ lack of convergence results [24], [25], since they 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. In NMF-based optimization problems, stationarity is an important property, since it guarantees that the reached limit point after a sequence of iterations corresponds to a local minimum. Moreover, as it has been shown in [25], update rules derived using projected gradients attain faster convergence compared to their multiplicative counterparts. In order to exploit these efficiencies we adopt such an optimization framework for the SDNMF problem. Using the cost function in (23) we formulate two subproblems, by keeping either $\mathbf{Z}$ or $\mathbf{H}$ fixed and performing optimization for the other. Consequently, two cost functions $\mathcal{O}_1(\mathbf{Z})$ and $\mathcal{O}_2(\mathbf{H})$ are derived where $\mathcal{O}_1(\mathbf{Z})$ assumes $\mathbf{H}$ is kept fixed, while $\mathcal{O}_2(\mathbf{H})$ assumes a fixed $\mathbf{Z}$:

$$\min_{\mathbf{Z}} \mathcal{O}_1(\mathbf{Z}) \quad \text{subject to:} \quad z_{i,k} \geq 0, \quad \forall i,k \tag{29}$$

$$\min_{\mathbf{H}} \mathcal{O}_2(\mathbf{H}) \quad \text{subject to:} \quad h_{k,j} \geq 0, \quad \forall k,j. \tag{30}$$

1) Optimization of $\mathbf{Z}$ *solving the subproblem (29):* The performed optimization is an iterative steepest descent process that at a given iteration round t the following update rule is applied:

$$\mathbf{Z}^{(t)} = P[\mathbf{Z}^{(t-1)} - \alpha_t \nabla \mathcal{O}_1(\mathbf{Z}^{(t-1)})], \tag{31}$$

$$h_{k,j}^{(t)} = h_{k,j}^{(t-1)} \frac{[\mathbf{Z}^{(t-1)^T}\mathbf{X}]_{k,j} + \beta[\mathbf{H}^{(t-1)} \sum_{r=1}^{n} \sum_{\theta=1}^{C_r} \frac{C-C_r}{N_{r,\theta}^2} \mathbf{e}_{r,\theta}^T \mathbf{e}_{r,\theta}]_{k,j}}{[\mathbf{Z}^{(t-1)^T}\mathbf{Z}^{(t-1)}\mathbf{H}^{(t-1)}]_{k,j} + \alpha[\mathbf{H}^{(t-1)}\mathbf{L}_w]_{k,j} + \beta \left[\mathbf{H}^{(t-1)}\text{diag}(\mathbf{e}) \left(\mathbf{1} - \sum_{r=1}^{C_r} \mathbf{e}_r^T \mathbf{e}_r\right) \text{diag}(\mathbf{e})\right]_{k,j}}, \quad (28)$$

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.

The determination of a proper learning step parameter α_t, at each iteration t, is crucial, since not only it determines convergence speed, but also constitutes a time consuming operation. An efficient approach for setting an appropriate value to parameter α_t based on the Armijo rule [35] is presented in [25], which is also adopted in this work. According to this strategy the learning step is computed as $\alpha_t = \beta^{g_t}$, where g_t is the first non-negative integer value found, such that the following inequality is satisfied:

$$\mathcal{O}_1(\mathbf{Z}^{(t)}) - \mathcal{O}_1(\mathbf{Z}^{(t-1)}) \leq \sigma\langle \nabla\mathcal{O}_1(\mathbf{Z}^{(t-1)}), \mathbf{Z}^{(t)} - \mathbf{Z}^{(t-1)}\rangle, \quad (32)$$

where operator $\langle ., . \rangle$ is the Frobenius inner product, which computes the sum of the entries of the Hadamard product between two matrices $\mathbf{A}$ and $\mathbf{B}$ as:

$$\langle \mathbf{A}, \mathbf{B}\rangle = \sum_{i,j} A_{i,j} B_{i,j} = \text{Tr}[\mathbf{A}^T\mathbf{B}]. \quad (33)$$

Parameters β and σ take values in range $(0,\ 1)$, while in our experiments we have set $\beta = 0.1$ and $\sigma = 0.01$ which is an efficient parameter selection, as has been verified in other studies [25], [36].

The cost function in (23) is quadratic in terms of $\mathbf{Z}$. Thus, $\mathcal{O}_1(\mathbf{Z})$ can be expanded near $\mathbf{Z}^{(t-1)}$ as follows:

$$\mathcal{O}_1(\mathbf{Z}^{(t)}) = \mathcal{O}_1(\mathbf{Z}^{(t-1)}) + \left(\mathbf{Z}^{(t)} - \mathbf{Z}^{(t-1)}\right)^T \nabla\mathcal{O}_1(\mathbf{Z}^{(t-1)}) \quad (34)$$
$$+ \frac{1}{2}\left(\mathbf{Z}^{(t)} - \mathbf{Z}^{(t-1)}\right)^T \nabla^2\mathcal{O}_1(\mathbf{Z}^{(t-1)}) \left(\mathbf{Z}^{(t)} - \mathbf{Z}^{(t-1)}\right).$$

By replacing (34) into (32), we derive the following inequality, which is less computationally expensive:

$$(1-\sigma)\left\langle \nabla\mathcal{O}_1(\mathbf{Z}^{(t-1)}), \mathbf{Z}^{(t)} - \mathbf{Z}^{(t-1)}\right\rangle \quad (35)$$
$$+ \frac{1}{2}\left\langle \mathbf{Z}^{(t)} - \mathbf{Z}^{(t-1)}, \nabla^2\mathcal{O}_1(\mathbf{Z}^{(t-1)})(\mathbf{Z}^{(t)} - \mathbf{Z}^{(t-1)})\right\rangle \leq 0.$$

By iterating the update rule in (31), 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 whether the currently reached limit point is stationary or not, a stationarity check step [36] is performed, which examines whether the following condition is satisfied:

$$||\nabla^P\mathcal{O}_1(\mathbf{Z}^{(t)})||_F \leq e_{\mathbf{Z}}||\nabla^P\mathcal{O}_1(\mathbf{Z}^{(1)})||_F, \quad (36)$$

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:

$$[\nabla^P\mathcal{O}_1(\mathbf{Z}^{(t)})]_{i,k} = \begin{cases} [\nabla\mathcal{O}_1(\mathbf{Z}^{(t)})]_{i,k} & \text{, if } z_{i,k} > 0 \\ \min\left(0, [\nabla\mathcal{O}_1(\mathbf{Z}^{(t)})]_{i,k}\right) & \text{, if } z_{i,k} = 0 \end{cases} \quad (37)$$

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

2) Optimization of $\mathbf{H}$ *solving the subproblem (30):* In order to find a stationary limit point for $\mathcal{O}_2(\mathbf{H})$, a similar procedure is followed. Initially, the learning step parameter α_t is determined and the weights matrix $\mathbf{H}$ is updated as follows:

$$\mathbf{H}^{(t)} = P[\mathbf{H}^{(t-1)} - \alpha_t\nabla\mathcal{O}_2(\mathbf{H}^{(t-1)})] \quad (38)$$

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:

$$(1-\sigma)\left\langle \nabla\mathcal{O}_2(\mathbf{H}^{(t-1)}), \mathbf{H}^{(t)} - \mathbf{H}^{(t-1)}\right\rangle \quad (39)$$
$$+ \frac{1}{2}\left\langle \mathbf{H}^{(t)} - \mathbf{H}^{(t-1)}, \nabla^2\mathcal{O}_2(\mathbf{H}^{(t-1)})(\mathbf{H}^{(t)} - \mathbf{H}^{(t-1)})\right\rangle \leq 0.$$

The update procedure is repeated, until the limit point of the sequence $\{\mathbf{H}^{(t)}\}_{t=1}^{\infty}$ becomes stationary. Consequently, similar to the stationarity control condition checked regarding the update of $\mathbf{Z}$, the following termination criterion is used:

$$||\nabla^P\mathcal{O}_2(\mathbf{H}^{(t)})||_F \leq e_{\mathbf{H}}||\nabla^P\mathcal{O}_2(\mathbf{H}^{(1)})||_F. \quad (40)$$

The presented strategy generates a sequence of minimizers $\{\mathbf{Z}^{(t)}, \mathbf{H}^{(t)}\}_{t=1}^{\infty}$ until the reached limit point is stationary.

The minimization of both subproblems in (29) and (30) 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:

$$\nabla\mathcal{O}_1(\mathbf{Z}) = \mathbf{Z}\mathbf{H}\mathbf{H}^T - \mathbf{X}\mathbf{H}^T \quad (41)$$
$$\nabla^2\mathcal{O}_1(\mathbf{Z}) = \mathbf{H}\mathbf{H}^T \quad (42)$$
$$\nabla\mathcal{O}_2(\mathbf{H}) = \mathbf{Z}^T\mathbf{Z}\mathbf{H} - \mathbf{Z}^T\mathbf{X} + \alpha\mathbf{H}\mathbf{L}_w - \beta\mathbf{H}\mathbf{L}_b \quad (43)$$
$$\nabla^2\mathcal{O}_2(\mathbf{H}) = \mathbf{Z}^T\mathbf{Z} \otimes \mathbf{I}_L + \alpha\mathbf{I}_M \otimes \mathbf{L}_w - \beta\mathbf{I}_M \otimes \mathbf{L}_b \quad (44)$$

where $\otimes$ denotes the Kronecker product operation. Consequently, inequality (39) 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:

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

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

E. Connections of SDNMF algorithm with the NPAF framework

In [13] a unified framework for various NMF-based methods has been proposed, that uses a fast gradient descent optimization algorithm. In order to exploit the merits of this unified framework we adapt our SDNMF algorithm to NPAF

by modifying appropriately the considered alignment matrix. More precisely NPAF framework that exploits the Kullback-Leibler (KL) divergence metric [38] considers the following optimization problem:

$$\mathcal{O}_{NPAF}(\mathbf{X}||\mathbf{ZH}) \triangleq \mathcal{O}_{KL}(\mathbf{X}||\mathbf{ZH}) + \frac{\alpha}{2}\mathrm{Tr}[\mathbf{HLH}^T], \quad (46)$$

where $\mathcal{O}_{KL}(\mathbf{X}||\mathbf{ZH})$ is the part of the NPAF cost function that measures the reconstruction error and $\mathbf{L}$ is a symmetric positive semidefinite patch alignment matrix, different for each specialized NMF-based algorithm. To unify SDNMF algorithm in the NPAF framework we can rewrite its cost function as follows:

$$\mathcal{O}^{KL}_{SDNMF}(\mathbf{X}||\mathbf{ZH}) = \mathcal{O}_{KL}(\mathbf{X}||\mathbf{ZH}) + \frac{\alpha}{2}\mathrm{Tr}[\mathbf{H}(\mathbf{L}_w - \frac{\beta}{\alpha}\mathbf{L}_b)\mathbf{H}^T], \quad (47)$$

which is equivalent to (46) with the alignment matrix $\mathbf{L}$ that encodes the discriminative information replaced by $\mathbf{L}_w - \frac{\beta}{\alpha}\mathbf{L}_b$. 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 [13] generative multiplicative or fast gradient descent update rules.

F. Subclass Discriminant Kernel NMF Algorithm (SDKNMF)

In order to model non-linearities in the extracted image features we derive the non-linear counterpart of the proposed algorithm called SDKNMF. Here 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 problem at hand can be summarized as follows: approximate a set of non-linear non-negative training sample vectors mapped on a polynomial feature space, using a linear combination of appropriately weighted non-linear non-negative basis vectors mapped on the same polynomial feature space in a discriminant manner.

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

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

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 PGS-DNMF 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, the expansion performed around the current solution estimate $\mathbf{Z}^{(t-1)}$ in (34), considering up to quadratic terms, is not valid for polynomial kernels of degree $d \geq 2$.

Similarly two subproblems are generated from (48) considering 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 (38) where the involved first and second order partial derivatives of the cost function with respect to $\mathbf{H}$, keeping variable $\mathbf{Z}$ fixed, are now evaluated as:

$$\nabla O_\phi\left(\mathbf{X}^\phi||\mathbf{Z}^\phi\mathbf{H}\right) = \mathbf{K}_{z,z}\mathbf{H} - \mathbf{K}_{z,x} + \alpha\mathbf{HL}_w - \beta\mathbf{HL}_b \quad (49)$$

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

The learning step parameter α_t is similarly determined using (45) and a stationarity condition check step is performed as in (40), 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 (31), while the optimal learning step parameter is now determined using (32) instead of (35), since the cost function for different Mercer's kernels is no longer quadratic in terms of $\mathbf{Z}$ and thus inequality (35) is not valid. Considering polynomial kernel functions of arbitrary degree the involved in (32) first order partial derivative with respect to $\mathbf{Z}$, is evaluated as:

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

As can be observed, all involved calculations can be performed using the so-called kernel trick. Thus, explicit computation of the mappings $\phi(\mathbf{z}_i)$ and $\phi(\mathbf{x}_j)$ is not required. Details regarding the derivation of the first order partial derivative with respect to $\mathbf{Z}$, when considering polynomial kernel functions for the non-linear mapping are available in Appendix B.

IV. Experimental Study

We compare the performance of the proposed methods, considering both optimization frameworks, with those of various NMF based algorithms, such as NMF, PGNMF [25], DNMF, PGKNMF [20], NDLA [13] and GNMF [11]. Moreover, we also include in our experimental comparison linear subspace learning methods such as CDA, LDA, PCA, LPP [39] and the Marginal Fisher Analysis (MFA) [40], 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 [41] and the Binghamton University 3D Facial Expression Database (BU-3DFE) [42], face recognition on the CMU-PIE dataset [43] and object recognition on the ETH-80 [44] image set. Figure 1 shows example images from the Cohn-Kanade dataset, depicting the seven recognized facial expression classes arranged in the following order: anger, fear, disgust, happiness, sadness, surprise and the neutral emotional state.

Fig. 1. Sample images depicting the different facial expressions from the Cohn-Kanade database.

A. Preprocessing of Facial Expression Data

To form our data collection, for the facial expression recognition experiments, 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 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 grayscale facial image was scanned row-wise, so as to form a feature vector which was used to compose either the training or the test set.

To measure the facial expression recognition accuracy, we randomly partitioned the available samples into 5-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 treated expression recognition problem, since identity related issues arose.

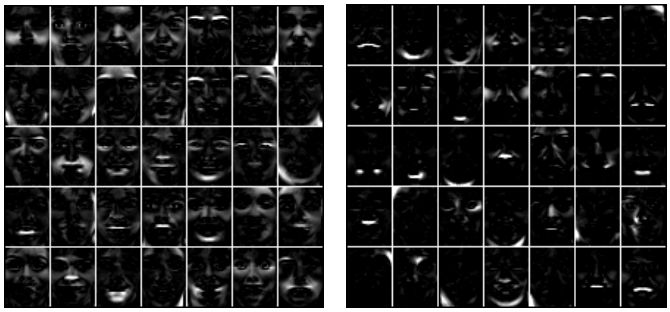
B. Cohn-Kanade dataset

The Cohn-Kanade AU-Coded facial expression database is among the most popular databases for benchmarking methods that perform facial expression recognition. Our data collection comprised of 407 images depicting 100 subjects, posing in 7 different emotional states (i.e. anger, disgust, fear, happiness, sadness, surprise and the neutral emotional state). As can be seen in Figure 1, Cohn-Kanade 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 partitioned into three subclasses and computed the mean expressive image for every cluster of each class according to the methodology presented in subsection III-C. Figure 2 shows the mean image for each facial expression considering the two more distant clusters inside each class. Clearly the illumination variations are captured during clustering.

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.

Table I summarizes the highest performance achieved by each examined method and the respective projection subspace dimensionality. All subclass discriminant algorithms (linear and non-linear) were found to attend an improved performance in this comparison. Moreover, the superiority of the projected gradients optimization framework is also demonstrated, since both PGNMF and PGSDNMF algorithms 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 learning algorithms 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 algorithm which also does not assume a Gaussian data distribution inside classes in order to enhance classes separability, in this experiment 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.

Figure 3 attempts a comparison between the basis images produced from training on the Cohn-Kanade database PGNMF and the proposed PGSDNMF algorithm, considering for the latter partitioning of each facial 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 the PGSDNMF algorithm. More precisely, PGSDNMF produced a few holistic basis images that contribute in minimizing the reconstruction error, while the majority are sparse and localized corresponding to specific local discriminant facial features, which participate in facial expression formation and lie mainly in the facial areas around mouth, eyes and eyebrows. These basis images significantly assist in facial expression discrimination and optimize the imposed discrimination criteria introduced in the PGSDNMF cost function. This observation reveals that the proposed method successfully decomposed each facial image into its discriminant facial features, a fact that verifies its superiority for the facial expression classification task.



(a) (b)

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

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)	$\mathbf{C_r = 2(200)}$

C. BU-3DFE Dataset

The dataset we generated from BU-3DFE contains 700 images, depicting 100 subjects performing 7 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 Cohn-Kanade 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 each captured under 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 4 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 [11]. We randomly selected half facial images of each individual for training, while the rest were used for testing. Table III summarizes the highest recognition rates achieved by each examined method. The experimental results showed that the proposed algorithms are more robust in 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 non-linear 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. Object Recognition on ETH-80 dataset

ETH-80 image dataset depicts 80 objects divided into 8 different classes, where for each object 41 images have been captured from different view points, spaced equally over the upper viewing hemisphere. Thus, the database contains 3,280 images in total. 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.

All algorithms applied on the same data matrix $\mathbf{X}^{16384\times 2000}$ and the best results are summarized in Table IV. As it can be seen, PCA outperformed all linear subspace learning algorithms, while SDKNMF considering five subclasses per each object class produced the best results among all examined methods. 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 recognition rate it attains for different parameter C_r values. As it can be seen in Figure 4 SDNMF efficacy initially increases as we partition each class from 2 up to 5 subclasses, where our algorithm attained its best performance, while after that point further partitioning classes results in reduced recognition accuracy. This is attributed to the fact that in these cases the training samples per subclass are limited and, consequently, the subclass covariance matrices are poorly estimated which affects the correctness of the identified projection directions [33], [46].

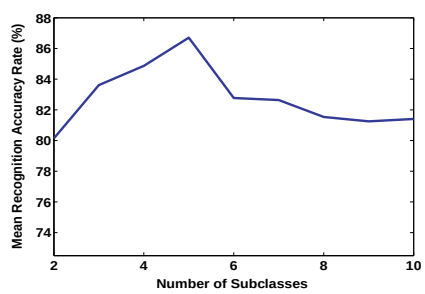


Fig. 4. Object recognition rate versus the number of subclasses each category of objects is partitioned to.

F. 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 $\mathbf{X} \in R_+^{1200\times 407}$ composed of all expressive images of the Cohn-Kanade dataset considering two subclasses partitioning of each expression class and setting the projection subspace dimensionality equal to 50. Moreover, parameters α and β influencing the contribution of the within and between subclass

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 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)	$\mathbf{C_r = 5(250)}$

scatter matrices trace optimization in the objective function were set for both algorithms to 0.5 and 0.9, respectively, while both algorithms were initialized using the same randomly generated matrices. Figure 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.

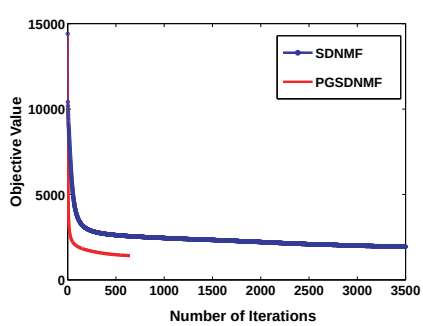


Fig. 5. Objective function value versus the number of iterations for the SDNMF and PGSDNMF algorithms.

To reveal the computational requirements of each method we measured the computational complexity per iteration for the derived update rules in (28) and (38) by counting the number of arithmetic operations required and summarized the results using the big O notation [47]. 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.

Based on the update rule in (28) for each iteration the overall cost for the SDNMF algorithm is $O(FLM)$. For PGSDNMF based on the alternative projective gradient approach and applying Algorithm 4 in [25] to determine properly the learning rate parameter α_t the complexity is $O(FLM^2) + t \times \big(O(rML^2)\big)$ where t is the number of iterations performed for the minimization of the subproblem in (30) 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 algorithm.

In Table V we show the recorded CPU training time, measured in seconds, required by NMF, PGNMF, SDNMF, PGSDNMF and PCA algorithms. All algorithms have been implemented on Matlab and the required by each method CPU time during training has been recorded. PCA required less training time than all NMF-based algorithms. This is attributed to the fact that both NMF and PGNMF are iterative optimization methods, and consequently are computationally expensive. In our implementation, we derived SDNMF from the original NMF algorithm modifying appropriately the multiplicative update rules and the stopping condition, while PGSDNMF has been devised from the PGNMF implementation provided by the authors of [25]. The difference in the training time between the PGSDNMF and PGNMF algorithms is attributed to the involved Kronecker product operation that significantly increases the size of the matrices involved in the computations of the first method.

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

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

G. 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. Since, we are interested in enhancing classes discrimination, thus increasing classification performance, we seek to determine these parameters with respect to the reached recognition accuracy. Thus, we seek for the C_r, α and β values that achieve the highest recognition rate.

More precisely, we performed a two stage cross validation process in order first to determine the optimal C_r value, while considering equal contribution of the discriminant factors setting $\alpha = \beta = 1$, and subsequently, to identify the optimal α and β values for that clustering setting. 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 is 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 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 Cohn-Kanade database we trained our algorithm considering $C_r = 2$, which was identified during the previous step and set values to parameters α and β in the range $[0, 1]$. Figure 6 shows the average reached expression recognition rates of SDNMF in Cohn-Kanade after 5 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 [16]. Thus, we applied $\alpha = 0.1$ and $\beta = 0.1$ on all experiments on facial image data, while on ETH-80 dataset setting $\alpha = 0.1$ and $\beta = 0.3$ yielded the best results.

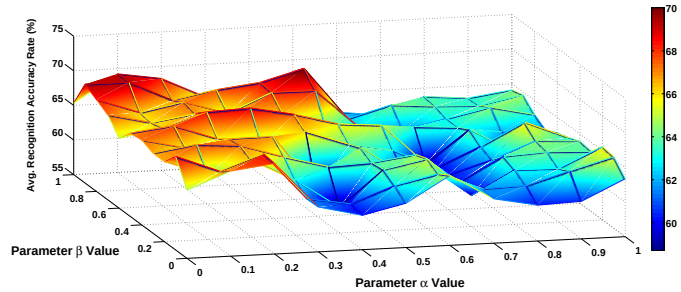


Fig. 6. Mean expression recognition rate in Cohn-Kanade database after five random starts of SDNMF algorithm versus the parameters α and β value.

V. CONCLUSIONS

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 which were incorporated 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 non-linear counterpart of the proposed method considering projections in non-linear polynomial feature spaces has been also investigated. We compared the performance of the proposed algorithms with that of various linear and non-linear competing methods for facial expression, face and object recognition verifying the effectiveness of the proposed methods in various recognition tasks.

APPENDIX A
PROOF OF CONVERGENCE

Theorem 1: *The objective function in (23) is non-increasing under the element-wise update rule in (28).*

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:

$$\mathbf{H}^{(t)} = \arg\min_{\mathbf{H}} G(\mathbf{H}, \mathbf{H}^{(t-1)}) \tag{52}$$

will never increase the objective function, since the following inequality is valid:

$$\begin{aligned} \mathcal{O}_{SDNMF}\left(\mathbf{H}^{(t)}\right) &\leq G\left(\mathbf{H}^{(t)}, \mathbf{H}^{(t-1)}\right) \leq \\ G\left(\mathbf{H}^{(t-1)}, \mathbf{H}^{(t-1)}\right) &= \mathcal{O}_{SDNMF}\left(\mathbf{H}^{(t-1)}\right). \end{aligned} \tag{53}$$

Lemma: The function in (54) is an auxiliary function for $F_{h_{k,j}}$, which is the part of (23) that is only relevant to $h_{k,j}$.

$$G(h,h_{k,j}^{(t-1)}) = F_{h_{k,j}}(h_{k,j}^{(t-1)}) + F'_{h_{k,j}}(h_{k,j}^{(t-1)})(h-h_{k,j}^{(t-1)}) + \frac{[\mathbf{Z}^T\mathbf{Z}\mathbf{H}]_{k,j} + \alpha[\mathbf{H}\mathbf{L}_w]_{k,j} + \beta[\mathbf{H}\text{diag}(\mathbf{e})(\mathbf{1}-\sum_{r=1}^{C_r}\mathbf{e}_r^T\mathbf{e}_r)\text{diag}(\mathbf{e})]_{k,j}}{2h_{k,j}^{(t-1)}}(h-h_{k,j}^{(t-1)})^2 \tag{54}$$

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:

$$F'_{h_{k,j}} = [\mathbf{Z}^T\mathbf{Z}\mathbf{H}]_{k,j} - [\mathbf{Z}^T\mathbf{X}]_{k,j} + \alpha[\mathbf{H}\mathbf{L}_w]_{k,j} - \beta[\mathbf{H}\mathbf{L}_b]_{k,j} \tag{55}$$

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

Obviously, according to the definition of the auxiliary function in (54) it holds: $G(h,h) = F_{h_{k,j}}(h)$. Consequently, we only need to show that $G(h,h_{k,j}^{(t-1)}) \geq F_{h_{k,j}}(h)$. In order 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:

$$F_{h_{k,j}}(h) = F_{h_{k,j}}(h_{k,j}^{(t-1)}) + F'_{h_{k,j}}(h-h_{k,j}^{(t-1)}) + \frac{1}{2}F''_{h_{k,j}}(h-h_{k,j}^{(t-1)})^2 \tag{57}$$

Substituting (56) into (57) and comparing it with (54), 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:

$$\frac{[\mathbf{Z}^T\mathbf{Z}\mathbf{H}]_{k,j} + \alpha[\mathbf{H}\mathbf{L}_w]_{k,j} + \beta[\mathbf{H}\text{diag}(\mathbf{e})(\mathbf{1}-\sum_{r=1}^{C_r}\mathbf{e}_r^T\mathbf{e}_r)\text{diag}(\mathbf{e})]_{k,j}}{h_{k,j}^{(t-1)}} \geq F_{h_{k,j}} \tag{58}$$

To prove the above inequality we will compare each term in (58) separately:

$$[\mathbf{Z}^T\mathbf{Z}\mathbf{H}]_{k,j} = \sum_{l=1}^{L}[\mathbf{Z}^T\mathbf{Z}]_{k,k}h_{k,l} \quad \geq \quad [\mathbf{Z}^T\mathbf{Z}]_{k,k}h_{k,j} \tag{59}$$

$$\alpha[\mathbf{H}\mathbf{L}_w]_{k,j} = \alpha\sum_{l=1}^{M}h_{k,l}[\mathbf{L}_w]_{l,j} \quad \geq \quad \alpha h_{k,j}[\mathbf{L}_w]_{j,j}. \tag{60}$$

To complete the proof we need to show that:

$$\begin{aligned}\left[\mathbf{H}\text{diag}(\mathbf{e})(\mathbf{1}-\sum_{r=1}^{C_r}\mathbf{e}_r^T\mathbf{e}_r)\text{diag}(\mathbf{e})\right]_{k,j} &\geq -h_{k,j}[\mathbf{L}_b]_{j,j} \Leftrightarrow \\ \left[\mathbf{H}\text{diag}(\mathbf{e})(\mathbf{1}-\sum_{r=1}^{C_r}\mathbf{e}_r^T\mathbf{e}_r)\text{diag}(\mathbf{e})\right]_{k,j} &\geq -h_{k,j} \\ \times\left[\sum_{r=1}^{n}\sum_{\theta=1}^{C_r}\frac{C-C_r}{N_{r,\theta}^2}\mathbf{e}_{r,\theta}^T\mathbf{e}_{r,\theta}\right]_{j,j}\end{aligned} \tag{61}$$

since

$$\left[\text{diag}(\mathbf{e})(\mathbf{1}-\sum_{r=1}^{C_r}\mathbf{e}_r^T\mathbf{e}_r)\text{diag}(\mathbf{e})\right]_{j,j} = 0 \tag{62}$$

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 (61) is simplified to:

$$\sum_{l=1}^{M}h_{k,l}\left[\text{diag}(\mathbf{e})(\mathbf{1}-\sum_{r=1}^{C_r}\mathbf{e}_r^T\mathbf{e}_r)\text{diag}(\mathbf{e})\right]_{l,j} + h_{k,j}\left[\sum_{r=1}^{n}\sum_{\theta=1}^{C_r}\frac{C-C_r}{N_{r,\theta}^2}\mathbf{e}_{r,\theta}^T\mathbf{e}_{r,\theta}\right]_{j,j} \geq 0 \tag{63}$$

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 $\left[\text{diag}(\mathbf{e})(\mathbf{1}-\sum_{r=1}^{C_r}\mathbf{e}_r^T\mathbf{e}_r)\text{diag}(\mathbf{e})\right]_{l,j} \geq 0$. Summing up all the above inequalities completes the proof.

Proof of Theorem 1: Substituting $G(h,h_{k,j}^{(t-1)})$ of (54) into (52) we derive:

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

Consequently, (54) is an auxiliary function of (23) and $\mathcal{O}_{SDNMF}$ is non-increasing under the update in (28).

Appendix B
First order partial derivatives with respect to $\mathbf{Z}$ considering arbitrary degree polynomial kernels

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

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

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

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

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

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

which in matrix form can be written as:

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

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, 1999.

[2] I. Jolliffe, *Principal component analysis*. New York: Springer-Verlag, 1986.

[3] A. Bell and T. Sejnowski, "The independent components of natural scenes are edge filters," *Vision research*, vol. 37, no. 23, pp. 3327–3338, 1997.

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

[5] G. Golub and C. Van Loan, *Matrix computations*, 3rd ed. Johns Hopkins, 1996.

[6] P. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.

[7] S. Li, X. Hou, H. Zhang, and Q. Cheng, "Learning Spatially Localized, Parts-based Representation," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001, pp. 207–212.

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

[9] C. Ding, T. Li, and M. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 45–55, January 2010.

[10] D. Cai, X. He, X. Wu, and J. Han, "Non-negative matrix factorization on manifold," in *IEEE International Conference on Data Mining (ICDM)*, 2008, pp. 63–72.

[11] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized non-negative matrix factorization for data representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1548–1560, August 2011.

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

[13] ——, "Non-negative patch alignment framework," *IEEE Transactions on Neural Networks*, vol. 22, no. 8, pp. 1218–1230, 2011.

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

[15] Y. Wang, Y. Jia, C. Hu, and M. Turk, "Non-negative matrix factorization framework for face recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 19, no. 4, pp. 495–511, 2005.

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

[17] I. Buciu and I. Pitas, "A new sparse image representation algorithm applied to facial expression recognition," in *MLSP*, Sao Luis, Brazil, Sept. 29 - Oct. 1 2004, pp. 539–548.

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

[19] I. Buciu, N. Nikolaidis, and I. Pitas, "Nonnegative matrix factorization in polynomial feature space," *IEEE Transactions on Neural Networks*, vol. 19, no. 6, pp. 1090–1100, 2008.

[20] S. Zafeiriou and M. Petrou, "Nonlinear non-negative component analysis algorithms," *IEEE Transactions on Image Processing*, vol. 19, no. 4, pp. 1050–1066, April 2010.

[21] K. Fukunaga, *Introduction to statistical pattern recognition*, 2nd ed. Academic Press, 1990.

[22] X. Chen and T. Huang, "Facial expression recognition: a clustering-based approach," *Pattern Recognition Letters*, vol. 24, no. 9-10, pp. 1295–1302, 2003.

[23] I. Buciu and I. Nafornita, "Non-negative matrix factorization methods for face recognition under extreme lighting variations," in *IEEE International Symposium on Signals, Circuits and Systems (ISSCS)*, 2009, pp. 1–4.

[24] C.-J. Lin, "On the convergence of multiplicative update algorithms for nonnegative matrix factorization," *IEEE Transactions on Neural Networks*, vol. 18, no. 6, pp. 1589–1596, November 2007.

[25] ——, "Projected gradient methods for nonnegative matrix factorization," *Neural Computation*, vol. 19, no. 10, pp. 2756–2779, 2007.

[26] S. Nikititidis, A. Tefas, and I. Pitas, "Using subclasses in discriminant non-negative subspace learning for facial expression recognition," in *19th European Signal Processing Conference (EUSIPCO 2011)*, Barcelona, Spain, August 29 - September 2 2011, pp. 1964–1968.

[27] S. Nikitidis, A. Tefas, N. Nikolaidis, and I. Pitas, "Subclass discriminant nonnegative matrix factorization for facial image analysis," *Pattern Recognition*, vol. 45, no. 12, pp. 4080–4091, 2012.

[28] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems (NIPS)*, 2000, pp. 556–562.

[29] R. Fletcher, *Practical methods of optimization; (2nd ed.)*. New York, NY, USA: Wiley-Interscience, 1987.

[30] Y. Xue, C. Tong, Y. Chen, and W. Chen, "Clustering-based initialization for non-negative matrix factorization," *Applied Mathematics and Computation*, vol. 205, no. 2, pp. 525–536, 2008.

[31] C. Boutsidis and E. Gallopoulos, "Svd based initialization: A head start for nonnegative matrix factorization," *Pattern Recognition*, vol. 41, no. 4, pp. 1350–1362, 2008.

[32] N. Guan, D. Tao, Z. Luo, and B. Yuan, "Nenmf: An optimal gradient method for non-negative matrix factorization," *IEEE Transactions on Signal Processing*, vol. 60, no. 6, pp. 2882–2898, 2012.

[33] M. Zhu and A. Martinez, "Subclass discriminant analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1274–1286, August 2006.

[34] A. Azran and Z. Ghahramani, "Spectral methods for automatic multi-scale data clustering," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006, pp. 190–197.

[35] D. Bertsekas, *Nonlinear programming*, 2nd ed. Athena Scientific, 1999.

[36] C. Lin and J. Moré, "Newton's method for large bound-constrained optimization problems," *SIAM Journal on Optimization*, vol. 9, no. 4, pp. 1100–1127, 1999.

[37] D. Bertsekas, "On the Goldstein-Levitin-Polyak gradient projection method," *IEEE Transactions on Automatic Control*, vol. 21, no. 2, pp. 174–184, April 1976.

[38] L. Bregman, "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming," *U.S.S.R. Computational Mathematics and Mathematical Physics*, vol. 1, pp. 200–217, 1967.

[39] X. He and P. Niyogi, "Locality preserving projections," in *Advances in Neural Information Processing Systems*, vol. 16, Vancouver, British Columbia, Canada, 2003.

[40] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 40–51, 2007.

[41] T. Kanade, J. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," in *IEEE International Conference on Automatic Face and Gesture Recognition*, March 2000, pp. 46–53.

[42] L. Yin, X. Wei, Y. Sun, J. Wang, and M. Rosato, "A 3D facial expression database for facial behavior research," *IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 211–216, April 2006.

[43] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression database," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1615–1618, 2003.

[44] B. Leibe and B. Schiele, "Analyzing appearance and contour based methods for object categorization," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2003.

[45] G. Bradski, A. Kaehler, and V. Pisarevsky, "Learning-based computer vision with intels open source computer vision library," *Intel Technology Journal*, vol. 9, no. 2, pp. 119–130, 2005.

[46] C. E. Thomaz, D. F. Gillies, and R. Q. Feitosa, "A new covariance estimate for bayesian classifiers in biometric recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 2, pp. 214–223, 2004.

[47] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. The MIT press, 2001.