Non-negative matrix factorization in polynomial feature space

I. Buciu [1,2] N. Nikolaidis [1] I. Pitas [1]

[1] Department of Informatics, Aristotle University of Thessaloniki

GR-541 24, Thessaloniki, Box 451, Greece

{pitas,nikolaid}@aiia.csd.auth.gr

[2] Department of Electronics, Faculty of Electrical Engineering and

Information Technology,University of Oradea

410087, Universitatii 1, Romania

ibuciu@uoradea.ro

Abstract

Plenty of methods have been proposed in order to discover latent variables (features) in data sets. Such approaches include the Principal Component Analysis (PCA), Independent Component Analysis (ICA), Factor Analysis (FA), etc., to mention only a few. A recently investigated approach to decompose a data set with a given dimensionality into a lower dimensional space is the so-called Non-negative Matrix Factorization (NMF). Its only requirement is that both decomposition factors are non-negative. To approximate the original data, the minimization of the NMF objective function is performed in the Euclidean space, where the difference between the original data and the factors can be minimized by employing L_2 norm. We propose a generalization of the NMF algorithm by translating the objective function into a Hilbert space (also called feature space) under nonnegativity constraints. With the help of kernel functions we developed an approach that allows high-order dependencies between the basis images while keeping the non-negativity constraints on both basis images and coefficients. Two practical applications, namely facial expression and face recognition, show the potential of the proposed approach.

Keywords

Image representation, feature extraction, kernel function, pattern recognition.

I. Introduction

Non-negative Matrix Factorization (NMF) decomposes a matrix $\mathbf{X}$ into two non-negative low rank matrices $\mathbf{W}$ (source matrix) and $\mathbf{A}$ (mixing matrix), such that $\mathbf{X} \approx \mathbf{WA}$ [1]. The idea of imposing nonnegativity constraints was partly motivated by the biological fact that the firing rates in visual perception neurons are non-negative. For instance, Hoyer [2] proposed to model the receptive fields using non-negativity sparse coding similar to NMF approach. Moreover, the non-negativity constraint arises in many real image processing applications. For example, the pixels in a grayscale image have non-negative intensities. In the case of NMF application to images, a set of n images $\mathbf{x}$ are lexicographically scanned and stored in the columns of the matrix $\mathbf{X}$. Then, as already mentioned, NMF decomposes $\mathbf{X}$ into a basis image matrix $\mathbf{W}$ and the corresponding coefficient matrix $\mathbf{A}$ [1]. NMF has been applied on a variety of applications, such as image classification [3], chemometry [4], sound recognition [5], musical audio separation [6] or extraction of summary excerpts from audio and video [7], air emission quality studies [8], identification of object materials from spectral reflectance data at different optical wavelengths [9], or text mining [10]. Two particular image processing tasks where NMF has been used are face and facial expression recognition. Regarding face recognition, the NMF approach has been compared to PCA and a variant of NMF, the so called Local Non-negative Matrix Factorization (LNMF) in [11]. LNMF has been proposed as a way to improve the NMF's basis image sparseness, as well as to reduce the redundant information between basis images in its decomposition. This is accomplished by imposing additional constraints related to spatial localization on the NMF associated cost function. Therefore, the localization of the learned image features is improved. The authors found that, while the NMF representation yields low recognition accuracy (actually lower than the one that can be obtained by using the PCA method), LNMF leads to a better classification performance for face recognition. In another paper [12] NMF and LNMF were compared on the task of face recognition on two face recognition databases, namely, YALE and AT&T. The results showed that these algorithms lead to quite different results, their performance being data dependent (i.e, for the YALE database NMF performed better than LNMF, while, for the AT&T database NMF was outperformed by LNMF). This could be due to differences in the database formation. Also, relatively poor performance was shown by NMF, when applied to facial expression recognition [13]. An NMF variant that allows sparse coding in both basis images and activity (its coefficients) was developed in [14]. The algorithm leads to an overcomplete decomposition, i.e., generates a larger number of basis vectors than the dimensionality of the input space.

Regardless of the input data, NMF is a linear model in the sense that an image is decomposed as a linear mixture of basis images. However, as suggested and evidenced by numerous works, the receptive fields exhibit nonlinear behavior [15], [16]. In other words, the response of the visual cells is a nonlinear function of their stimuli, where the response is characterized and analyzed on a low dimensional subspace [17], [18]. On the other hand, it was recently argued and proved that, in order to achieve an efficient perceptual coding system, a nonlinear image representation should be developed [19]. As described in that paper, employing an adaptive nonlinear image representation algorithm results in a reduction of the statistical and the perceptual redundancy amongst representation elements. As far as the pattern recognition (and, in particular, the face and facial expression recognition) task is concerned, the underlying features most useful for class discrimination may lie within the higher order statistical dependencies among input features. For example, Bartlett et al. [20] have demonstrated that the ICA is superior to PCA in human face recognition in that ICA learns the higher-order dependencies in the input besides the correlations. However, whether the facial expression is composed of a set of independent components is not clear yet. The aspects described above bring arguments in favor of developing a nonlinear counterpart of the NMF. Therefore, the aim of a nonlinear NMF variant is twofold: (a) to yield a model compatible with the neurophysiology paradigms (non-negativity constraints and nonlinear image decomposition) and (b) to discover higher-order correlation between image pixels that lead to more powerful (in discriminative terms) latent features.

One way to handle nonlinear correlation can be provided by using kernel theory. Kernel-based subspace methods have been extensively investigated in the literature. Nonlinear methods based on the kernel theory, such as Kernel PCA and Kernel Fisher Linear Discriminant were used for face recognition or denoising purposes and they were found to outperform their linear variants [21]. In [22] kernels are decomposed in order to obtain posterior probabilities for the class membership in a data clustering application. The kernel theory was pushed further and was applied for retrieving independent features from a non-linear mixture of sources. This has led to a kernel-based Independent Component

Analysis proposed by Bach and Jordan [23]. Hyperkernels have been introduced in [24], where the kernel is defined on the space of kernels itself, an approach which allows the adaptation of the kernel function instead of its parameters. An efficient way to adapt such hyperkernels using second-order cone programming is described in [25]. Recently, a combination of kernel theory and Fisher Linear Discriminant criterion has been proposed in [26] to extract the most discriminant nonlinear features and select a suitable kernel simultaneously.

In the current paper, we make use of the kernel functions to develop a decomposition method, where the discovered features (encompassed by the basis images) posses non-linear dependencies, while the decomposition factors remain non-negative. In the light of the kernel theory, we come up with a new formulation of NMF, where, although the decomposition is still linear, the discovered features have non-linear dependencies. Here, the nonlinearity aspect refers only to the relation between the pixels of basis images. In principal, the original data residing in a given space are firstly transformed by a nonlinear polynomial kernel mapping into a higher dimensional space, the so called reproducing kernel Hilbert space (RKHS) and then a nonnegative decomposition is accomplished in the feature space. The nonlinear mapping enables implicit characterization of the data high-order statistics. By using a polynomial kernel function, the basis image features are higher-order correlated, as we shall demonstrate in subsequent sections. We call the proposed approach the Polynomial kernel Non-negative Matrix Factorization (PNMF).

Another important issue appears when the samples from the database are recorded under varying lighting conditions which can cause the linear approach to perform poorly [27]. It is known that when the Lambertian assumption regarding the illumination is violated (i.e., the change in illumination is drastic) the linear subspace methods may fail. Although we did not ran systematic experiments that involve an in-depth analysis of the PNMF performance in the case of illumination changes, we can report preliminary results on a database containing samples recorded under varying lighting conditions, where PNMF outperformed other methods.

The remainder of the paper is organized as follows. The mathematical description of the NMF approach along with the corresponding cost function and its minimization is given in Section II. The Polynomial kernel Non-negative Matrix Factorization is developed in Section III. The potential of the method is investigated in the case of the face and facial expression recognition tasks. The data description and the experimental setup are presented in Section IV. The experimental results obtained by the PNMF algorithm are further compared with those of NMF, LNMF, PCA and ICA in Section IV. Also, as PNMF is a generalization of the NMF algorithm developed on the basis of kernel theory, its performance is compared against the nonlinear counterparts of PCA and ICA, namely kernel PCA (KPCA) and kernel ICA (KICA). Conclusions are drawn in Section V.

II. Non-negative matrix factorization

Suppose that we have a data space $\mathcal{X}$ and n non-negative input (training) m - dimensional vectors $\mathbf{x}_j = [x_{1j}, x_{2j}, \ldots, x_{ij}, \ldots, x_m]^T \in \mathcal{X}$ stored in a matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$, $i = 1, \ldots, m$, $j = 1, \ldots, n$. Suppose, also, that we can approximate the input data through a linear combination of a smaller set $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_p]$, with $\mathbf{w}_r = [w_{1r}, w_{2r}, \ldots, w_{ir}, \ldots, w_{mr}]^T \in \mathcal{X}$ (called basis images), $r = 1, \ldots, p$, and $p < n$. That is, each column $\mathbf{x}_j$ of $\mathbf{X}$ can be written as linear combination of this set, i.e. $\mathbf{x}_j \approx \mathbf{W}\mathbf{a}_j$, where $\mathbf{a}_j$ is a $p \times 1$ vector containing the linear decomposition coefficients. *Non-negative matrix factorization* finds $\mathbf{W}$ and $\mathbf{a}_j$ in a such a way that both the decomposition matrix $\mathbf{W}$ and coefficients $\mathbf{a}_j$ contain non-negative elements (i.e. $W_{ir}, a_r \geq 0$) [1]. Expanding the approximation to all data we have $\mathbf{X} \approx \mathbf{W}\mathbf{A}$. The quality of approximation depends on the cost function used. Two cost functions were proposed by Lee and Seung: the squared Euclidean distance between $\mathbf{X}$ and $\mathbf{W}\mathbf{A}$ and the Kullback-Leibler divergence [28], [29]. The squared Euclidean distance $\|\mathbf{X} - \mathbf{W}\mathbf{A}\|^2$ can be minimized via Expectation Maximization (EM)

[30], leading to the following iterative algorithm for updating the decomposition factors $\mathbf{A}$ and $\mathbf{W}$, at each iteration t [28]:

$$\mathbf{A}^{(t)} = \mathbf{A}^{(t-1)} \otimes (\mathbf{W}^{(t-1)T}\mathbf{X}) \oslash (\mathbf{W}^{(t-1)T}\mathbf{W}^{(t-1)}\mathbf{A}^{(t-1)}) \tag{1}$$

$$\mathbf{W}^{(t)} = \mathbf{W}^{(t-1)} \otimes (\mathbf{X}\mathbf{A}^{(t-1)T}) \oslash (\mathbf{W}^{(t-1)}\mathbf{A}^{(t-1)}\mathbf{A}^{(t-1)T}) \tag{2}$$

$$\mathbf{W}^{(t)} = \mathbf{W}^{(t)} \oslash \mathbf{S} \tag{3}$$

where $\otimes$ and $\oslash$ denote elementwise multiplication and division, respectively and T denotes matrix transposition. Equation (3) normalizes the basis images such that $w \in [0, 1]$, i.e. $\mathbf{S}$ is a $m \times p$ matrix, whose columns are given by $\mathbf{s}_r = \sum_{i=1}^{m} W_{ir}$, $r = 1, \ldots, p$.

Regarding NMF variant whose cost function is based on Kullback-Leibler divergence, it can be shown to be a particular version of Bregman divergence [31]. A general NMF decomposition expression based on Bregman divergence is modeled in [32] to encompass various special NMF algorithm variants derived in the literature.

III. Non-negative matrix factorization in polynomial feature space

Before deriving a non-negative matrix factorization in a polynomial feature space, we give the following two definitions:

Definition 1: A *kernel* is a function κ that satisfies $\kappa(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle$, for all $\mathbf{x}, \mathbf{z} \in \mathcal{X}$, where ϕ is a mapping from $\mathcal{X}$ to an (inner product) feature space $\mathcal{F}$, $\phi : \mathbf{x} \longrightarrow \phi(\mathbf{x}) \in \mathcal{F}$ [33].

Here $\langle ., . \rangle$ denotes the inner product.

Definition 2: Given two matrices $\mathbf{X}$ and $\mathbf{Y}$ of dimensions $m \times n$ and $m \times p$, respectively, the *kernel matrix* $\mathbf{K} \in \mathcal{X}^{n \times p}$ has elements $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{y}_j)$ for the data $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathcal{X}$, $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_p \in \mathcal{X}$ and some kernel function k.

To have a first idea about the role of the kernel function let us consider an example of a two-dimensional input space $\mathbf{x} = (x_1, x_2) \in \mathcal{X}^{2 \times 1} \subseteq \mathbb{R}^2$ together with the feature map

$\phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2) \in \mathcal{F} \subseteq \mathbb{R}^3$ [33]. The space of linear functions in $\mathcal{F}$ would be of the form:

$$g(\mathbf{x}) = \alpha_{11}x_1^2 + \alpha_{22}x_2^2 + \alpha_{12}\sqrt{2}x_1x_2. \tag{4}$$

As one can see, the feature map maps the data from a two dimensional to a three dimensional space in such way that the linear relations in the feature space correspond to quadratic relations in the input space. The use of kernel functions eliminates the need for an explicit definition of the nonlinear mapping Φ, because the data appear in the feature space only as dot products of their mappings:

$$\begin{aligned}\langle\phi(\mathbf{x}), \phi(\mathbf{z})\rangle &= \langle(x_1^2, x_2^2, \sqrt{2}x_1x_2), (z_1^2, z_2^2, \sqrt{2}z_1z_2)\rangle \\ &= x_1^2z_1^2 + x_2^2z_2^2 + 2x_1x_2z_1z_2 \\ &= (x_1z_1 + x_2z_2)^2 \\ &= \langle\mathbf{x}, \mathbf{z}\rangle^2.\end{aligned} \tag{5}$$

Frequently used kernel functions are the polynomial ones, $\kappa(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)^d$, and the Gaussian ones, $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/(2\sigma^2))$. This paper deals with the polynomial kernels.

Let us assume now that our input data $\mathbf{X} \in \mathcal{X} \subseteq \mathbb{R}^{m\times n}$ are transformed to the higher dimensional space $\mathcal{F} \subseteq \mathbb{R}^{l\times n}$, $l \gg m$. We denote the set of the transformed input data with $\mathbf{F} = [\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \ldots, \phi(\mathbf{x}_n)]$, where the l - dimensional vector $\phi(\mathbf{x}_j) = [\phi(\mathbf{x})_1, \phi(\mathbf{x})_2, \ldots, \phi(\mathbf{x})_s, \ldots, \phi(\mathbf{x})_l]^T \in \mathcal{F}$. We can find a matrix $\mathbf{Y} = [\phi(\mathbf{z}_1), \phi(\mathbf{z}_2), \ldots, \phi(\mathbf{z}_p)]$, $\mathbf{Y} \in \mathcal{F}$, that approximates the transformed data set, such that $p < n$. Therefore, each vector $\phi(\mathbf{x})$ can be written as a linear combination as $\phi(\mathbf{x}) \approx \mathbf{Yb}$. We introduce the following squared Euclidean distance in the space $\mathcal{F}$ between the mapping of the vector $\mathbf{x}_j$ and its decomposition factors as being our cost function:

$$q_j = \frac{1}{2}\|\phi(\mathbf{x}_j) - \mathbf{Yb}_j\|^2. \tag{6}$$

The aim is now to minimize q_j subject to $b_r, Z_{ir} \geq 0$, and $\sum_{i=1}^{m} Z_{ir} = 1$.

Theorem III.1: For the polynomial kernels of degree d, $\kappa(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)^d$ the cost function $Q = \|\phi(\mathbf{X}) - \mathbf{YB}\|^2$ is non-increasing under the following updating rules, for each iteration t:

$$\mathbf{B}^{(t)} \Leftarrow \mathbf{B}^{(t-1)} \otimes \mathbf{K}_{zx}^{(t-1)} \oslash (\mathbf{K}_{zz}^{(t-1)} \mathbf{B}^{(t-1)}) \tag{7}$$

$$\mathbf{Z}^{(t)} \Leftarrow \mathbf{Z}^{(t-1)} \otimes [(\mathbf{X}\mathbf{K}_{xz}^{'(t-1)}) \oslash (\mathbf{Z}^{(t-1)} \mathbf{\Omega} \mathbf{K}_{zz}^{'(t-1)})] \tag{8}$$

$$\mathbf{Z}^{t} \Leftarrow \mathbf{Z}^{t} \oslash \mathbf{S} \tag{9}$$

where $\mathbf{K}_{zx} := \langle \phi(\mathbf{z}_i), \phi(\mathbf{x}_i) \rangle$ and $\mathbf{K}_{xz} := \langle \phi(\mathbf{x}_i), \phi(\mathbf{z}_i) \rangle$ are kernel matrices of dimensions $p \times n$ and $n \times p$, respectively, containing values of kernel functions of $\mathbf{z}_i \in \mathbf{Z}$ and $\mathbf{x}_i \in \mathbf{X}$, and $\mathbf{K}_{zz} = \langle \phi(\mathbf{z}_i), \phi(\mathbf{z}_o) \rangle$ is a $p \times p$ kernel matrix of any vectors $\mathbf{z}_i, \mathbf{z}_o \in \mathbf{Z}$. $\mathbf{\Omega}$ is a diagonal matrix whose diagonal elements are $\omega_{rr} = \sum_{j=1}^{n} B_{rj}$, $r = 1, \ldots, p$. The columns of $\mathbf{S}$ are given by $\mathbf{s}_r = \sum_{i=1}^{m} Z_{ir}$, $r = 1, \ldots, p$.

The proof of (7) and (8) is given in Appendix. The sign " $'$ " denotes the derivative of matrix elements (functions). For the polynomial kernel $k^{'}(\mathbf{x}_i, \mathbf{x}_j) = d \cdot k(\mathbf{x}_i \cdot \mathbf{x}_j)^{d-1}$. Note that, if the non-negativity constraint is not imposed in the decomposition coefficients, then, the coefficients can be computed as (see equation (A-10) in Appendix):

$$\mathbf{B} = (\mathbf{K}_{zz})^{-1} \mathbf{K}_{zx}. \tag{10}$$

The choice of the Euclidean distance as a cost function for the non-linear feature space was motivated by the fact that we want to avoid to explicitly express the nonlinear mapping $\phi(\mathbf{x})$ and $\phi(\mathbf{z})$. Indeed, if we expand equation (6), taking into account equation (5), we have:

$$Q = \mathbf{k}(\mathbf{x}, \mathbf{x}) - 2\mathbf{k}(\mathbf{x}, \mathbf{z}_i)\mathbf{b} + \mathbf{b}^T \mathbf{k}(\mathbf{z}_i, \mathbf{z}_o)\mathbf{b}, \tag{11}$$

In other words, the problem can be easily solved by invoking only the kernel function. The polynomial kernel corresponds to an inner product in the space of d - th order monomials of the input space. If $\mathbf{x}$ represents an image, then, we work in the feature space which is spanned by the products of any d pixels. If we would need to work with the mapped value $\phi(\mathbf{x})$, the dimensionality would be, for example, $l = 10^{10}$ for a 16×16 pixels image and $d = 5$. Thus, by using polynomial kernel we can take into account higher-order image statistics without being concerned about the "curse of dimensionality". The PNMF's complexity is $O(mnpd)$ compared to $O(mnp)$ corresponding to NMF. Unfortunately, the updating scheme does not guarantee a global minimum due to its nonconvex optimization structure (applied simultaneously to B and Z), but only a local minimum. The local minimum that is reached depends on the initialization, i.e. the initial values of B and Z, usually chosen randomly. PNMF algorithm suffers from the same optimization drawback as NMF. One way to partially overcome this problem, i.e. prevent the algorithm from getting "stuck" in a "shallow" local minimum is to run it several times with different initializations.

It should be noted that the way the development of the iterative approach for updating the decomposition factors was carried out does not permit a non-negative decomposition for a RBF kernel. This is due to the negative solution resulting from the derivative associated to the RBF kernel. Other approaches have to be found for allowing a more flexible kernel.

The developed algorithm is closely related to the reduced set methods applied to Support Vector Machines (SVMs) [34], [35]. These approaches were developed in order to increase the speed of the SVMs and to reduce the computational complexity of kernel expansion by approximating them by using fewer terms without a significant loss in accuracy. The same Euclidean cost function was used, but no non-negativity constraint was imposed on the computation of the reduced set and their coefficients. Also, the input data of the reduced set method comes from the SVM output, which is a decision function depending on the Lagrangian computed by the SVMs optimization procedure and the kernel formed from the training and test data, while, in our case the input is formed only from the non-linear mapping of the original data.

IV. Experimental performance and evaluation setup

To asses the performance of the PNMF method, experiments on face and facial expression recognition were conducted. For comparison purpose NMF and LNMF have been involved in the experiments. Also, ICA [36] and PCA [37], along with their nonlinear variants namely kernel ICA (KICA) [38] and kernel PCA (KPCA) [39], respectively, were used.

A. Data description and processing

Two databases were used for facial expression recognition and another two databases were used for the face recognition task. The first set of facial images used for the facial expression recognition task come from the Cohn-Kanade AU-coded facial expression database [40]. The database was originally created for the representation of Action Units (AU) appearing in the FACS coding system and not for explicit facial expression recognition. The facial actions (action units) that are described in the image annotations have been converted into facial expression class labels according to [41]. Despite the fact that 100 posers were available, we were only able to identify thirteen of them who displayed all six facial basic expressions, namely anger, disgust, fear, happiness, sadness and surprise. These thirteen persons have been chosen to create the image database that has been used in our expression recognition experiments. Each subject from Cohn-Kanade database forms an expression over time starting from a neutral pose and ending with a very intense expression, thus having several video frames with different expression intensities. However, the number of these intermediate video frames is not the same for the various posers. We have selected three poses with low (close to neutral), medium and high (close to the maximum) facial expression intensity, as depicted in Figure 1, and used them to form the database utilized in our experiments.

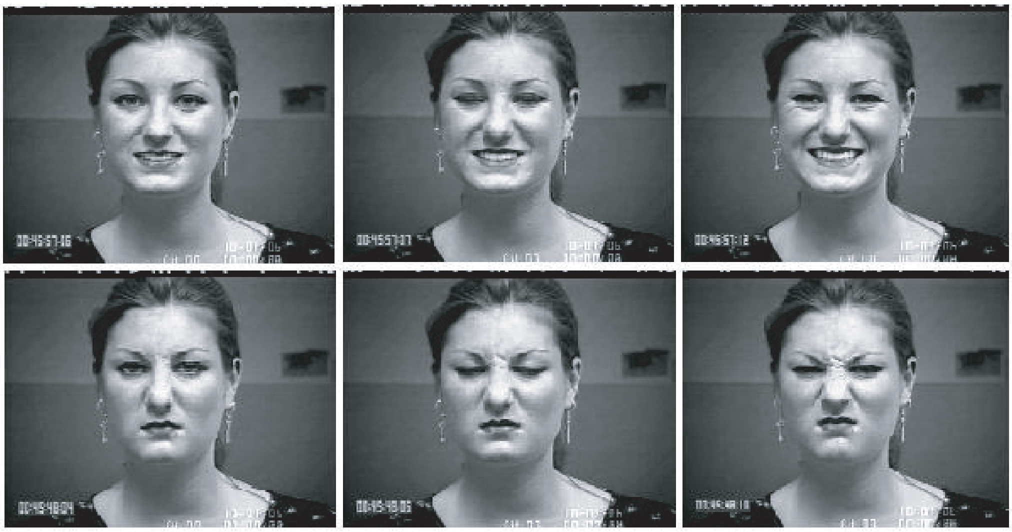

Fig. 1. Frames corresponding to the three selected intensities (low, medium and high) for the happiness and disgust expression as expressed by the same subject.

The total number of frontal images used was 234. The registration of each original frontal image $\mathbf{x}$ was performed by mouse clicking on the eyes, thus retrieving the eyes coordinates, followed by an image shift step for centering the eyes. Furthermore, the images were rotated to align the face horizontally according to the eyes. In the next step, the face region was cropped in order to remove the image borders, while keeping the main facial features (as eyebrows, eyes, nose and chin). Finally, each resulting image of size 80×60 pixels was downsampled to a final size of 40×30 pixels for computational load reduction. The face image pixels were stored into a 1200 - dimensional vector ($m = 1200$). These vectors formed the columns of matrix $\mathbf{X}$.

The second database contains 213 images of Japanese female facial expressions (JAFFE) [42]. Ten subjects produced 3 or 4 examples of each of the 6 basic facial expressions plus a neutral pose, thus producing a total of 213 images of facial expressions. Image registration was performed in the same way as for the Cohn-Kanade database.

As far as the face recognition task is concerned, two other databases were used. The first one, Yale face database [43] contains 165 grayscale images of 15 individuals. There are 11 images per subject, one per different facial expression or configuration: center-light, with glasses, happy, left-light, without glasses, normal, right-light, sad, sleepy, surprised, and winking. The second database was the AT&T face database [44] that contains ten different images for forty distinct subjects. All images have been shot against a dark homogeneous background with the subjects in an upright, frontal position with tolerance for slight pose differences. For computational reasons the image size was reduced to 42×31 pixels in the case of Yale and to 42×34 pixels for the AT&T database, respectively, after being preprocessed in order to align them using the previous mentioned procedure for the Cohn-Kanade database. It should be noted that, in the case of the face recognition databases, we did not crop the original images since the whole face together with the hair part could contain important information about a person's identity.

B. Training procedure

Before applying the training phase which consists in running the algorithm until it converges and retrieving the basis images and the coefficients, the n face images were split into a training set containing n_{tr} images and a disjoint test set containing n_{te} images. The corresponding matrices were $\mathbf{X}_{tr}$ and $\mathbf{X}_{te}$, respectively. The training images $\mathbf{X}_{tr}$ are used in the expressions (7) - (9) for updating $\mathbf{B}$ and $\mathbf{Z}$. To form the training set, $n_{tr} = 164$ and $n_{tr} = 150$ face images were randomly chosen from the Cohn-Kanade derived and the JAFFE database, respectively, while the remaining $n_{te} = 70$ and $n_{te} = 63$ images were used for testing, thus forming the test image set. Both the training and the test set contained all facial expressions. Finally, in the case of face recognition, for the Yale database, the first six images of each subject were used to form the training data set, while the remaining five samples were used as test images. In the case of the AT&T database, the images have been randomly split in 200 training images and 200 test images. For NMF, LNMF and PNMF, each training face image $\mathbf{x}_{tr}$ was projected by using the pseudoinverse basis images matrix, resulting in a feature vector $\mathbf{f}_{tr} = \mathbf{Z}^{-1}\mathbf{x}_{tr}$, where $\mathbf{x}_{tr}$ is modified to become zero mean. For the ICA approach, which has been used for comparison, we used the first architecture described in [20] that gives us the coefficients to be applied for classification. The ICA decomposition coefficients of each image form essentially a row of the matrix $\mathbf{F}_{tr} = \mathbf{X}_{tr}\mathbf{P}_p\mathbf{A}^{-1}$. Here $\mathbf{P}_p$ is the projection matrix resulting from PCA procedure applied prior to ICA and $\mathbf{A}$ is the unmixing matrix found by the ICA algorithm. The number of independent components is controlled by the first p eigenvectors [20]. PCA alone was also applied on our experimental data. In this case, the eigenvectors (eigenimages) are extracted from the database comprising the training images. Further, the training images are projected onto the transpose of the matrix which contains the eigenimages, thus yielding the training feature vector corresponding to the PCA approach. The same strategy was adopted for KICA and KPCA. However, we must notice that, in order to have a fair comparison to PNMF, KPCA was used only with the polynomial kernel in our experiments.

C. Test procedure

In the test phase, and for NMF, LNMF and PNMF, each test face image $\mathbf{x}_{te}$ was modified so as to become zero mean one and subsequently, a test feature vector $\mathbf{f}_{te}$ was formed by $\mathbf{f}_{te} = \mathbf{Z}^{-1}\mathbf{x}_{te}$. For the ICA approach, the test feature vector was formed as $\mathbf{f}_{te} = \mathbf{x}_{te}\mathbf{P}_p\mathbf{A}^{-1}$. For the PCA, the test images were projected onto the eigenimages achieved in the training step, yielding the PCA test feature vectors. The same strategy was adopted for KICA and KPCA.

D. Classification procedure

The six basic facial expressions (i.e. *anger*, *disgust*, *fear*, *happiness*, *sadness* and *surprise*) available for the Cohn-Kanade database form six classes. One more class (*neutral*) exists in the case of the JAFFE database. Regarding the YALE and AT&T database that were used in the face recognition experiment, each class represents an individual and the class labels denote the identity of each individual. If we construct a classifier, whose class label output for a test sample $\mathbf{f}_{te}$ is $\widetilde{l}$, then the classifier accuracy is defined as the percentage of the correctly classified test images, i.e., the percentage of images $\widetilde{l}(\mathbf{f}_{te}) = l(\mathbf{f}_{te})$, where $l(\mathbf{f}_{te})$ being the correct class label. Three classifiers were employed for classifying the features extracted by the algorithms. The first classifier is a nearest neighbor classifier based on the cosine similarity measure (CSM). This approach uses as similarity measure the cosine of angle between a test feature vector and a prototype one, i.e. one derived from the training phase. More specifically, $\widetilde{l} = l(\mathbf{f}_{k,tr})$ where $k = \text{argmin}_{i=1,\dots,n_{tr}}\{d_i\}$ and $d_i = \frac{(\mathbf{f}_{te})^T\mathbf{f}_{i,tr}}{\|\mathbf{f}_{te}\|\|\mathbf{f}_{i,tr}\|}$. The second classifier is a two layer neural network (RBFNN) based on radial basis functions (RBFs) $g(x) = \exp(-\|\mathbf{f}_i - \mathbf{f}_j\|^2/(2\sigma^2))$, where $\mathbf{f}$ is the feature vector associated to either training or test image. Finally, the third classifier is based on SVMs [45] with different kernels (linear, polynomial, and RBF). The sequential minimal optimization technique developed by Platt [46] was used to train the SVMs. Since classical SVM theory was intended to solve a two class classification problem, we chose the Decision Directed Acyclic Graph (DDAG) learning architecture proposed by Platt et al. to cope with the multi-class classification [47].

E. Performance evaluation and discussions

E.1 Facial expression recognition

We ran the PNMF algorithm for various number of basis images p and different values of the polynomial degree $d = \{2,3,4,5,6,7,8,9,10\}$. Also, the same polynomial degree range was used for KPCA and the results presented are the ones that correspond to the degree that gave the best results. Several basis images discovered by PNMF are depicted in Figure 2 for the Cohn-Kanade database and for different values of d. The basis images corresponding to $d = 2$ and $d = 3$ are noisy. As the degree increases more pixels are taken into account. This leads to a "finer" image representation and an emphasis on the expression. The phenomenon can be easily observed especially in the third basis image of Figure 2, where the happiness expression passes through different intensities (from a vague "smile" to an intense "smile").

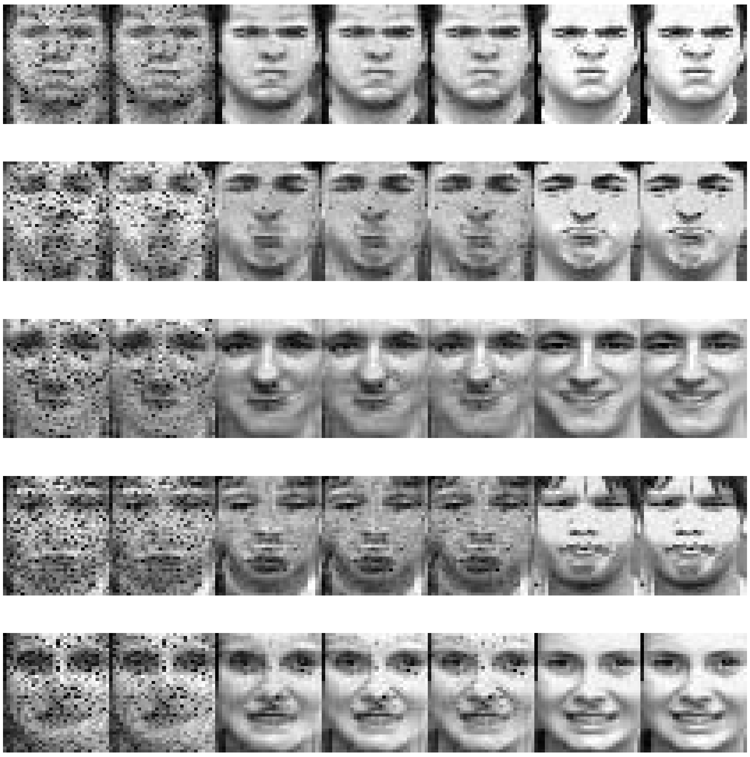

Fig. 2. Five different basis images retrieved by the PNMF with $d = \{2,3,4,5,6,7,8\}$ (left to right) for the Cohn-Kanade database.

The classification results for the facial expression recognition task for different image representations and classifiers (CSM, RBFNN, SVM) involved in the experiment are shown in Table I. The minimum number of basis images p corresponding to the maximum classification accuracy is also tabulated. The results of the six other subspace image representations (NMF, LNMF, PCA, KPCA, ICA and KICA) are also presented. For all three databases and all classifiers, PNMF outperformed all other methods. Generally, for both Cohn-Kanade (C-K) and JAFFE databases, the best results are provided when the features are classified by SVM followed by CSM and RBFNN. As far as the feature extraction algorithm is concerned, in the case of C-K, the best classification performance was achieved by PNMF, while the second best performance was attributed to the LNMF approach. A greater difference in performance between the best (PNMF) and the second best algorithm (NMF and LNMF) was obtained for the JAFFE database, where PNMF outperforms NMF by almost 3 %, in the case of SVM classifier. Both KPCA and KICA have shown superior performance compared to PCA and ICA, respectively. However, they performed worse than PNMF. Interestingly, KPCA and KICA achieved lower accuracy than PCA and ICA when they classified facial expressions from the C-K database. This is in line with the results reported in [48] where KPCA was found inferior in performance to PCA for the JAFFE database. Compared with the Cohn-Kanade database, the JAFFE database leads to lower classification performance, due to the fact that the subjects posing for this database are not as expressive as those in the Cohn-Kanade, making facial expression harder to be recognized. As experiments showed, the difference between the classification performance of the PNMF (best one) and the second best one is larger in the case of the JAFFE database than in the Cohn-Kanade database. This fact is an indication that the benefit from using PNMF is more prominent in cases where classes are difficult to separate.

It has been argued [49], [13] that, by performing the processing on difference images obtained by subtracting each expression image from its corresponding neutral pose, when available, the classification accuracy is much improved. Thus an experiment involving difference images was conducted. The difference images were formed for the JAFFE database and the new database was denoted by JAFFE$_{diff}$. The same procedure as above was then applied on the new database. Indeed, the accuracy increased for all image representation approaches and all classifiers. An impressive gain was achieved in the case of the PNMF with CSM, where the accuracy increased from 69.8% in JAFFE up to 93.8% in JAFFE$_{diff}$. In terms of classifiers the highest accuracy is obtained by CSM followed by NN and SVM. However, regardless of classifier used, again, PNMF performed better than all other approaches, including KPCA and KICA. A slight accuracy improvement was observed for the KPCA over PCA.

E.2 Face recognition

The second experiment dealt with face identity recognition. Several basis images from the YALE database retrieved by the PNMF algorithm for $d = \{2,3,4,5,6,7,8,9,10\}$ are depicted in Figure 3. Notice that for the first, fourth and fifth basis image of this Figure, as the degree increases the same subject is represented, whilst in the third different subjects are represented. For the second one, the basis image corresponding to $d = 6$ and $d = 7$ is a mixture of two subjects (the one mainly represented by the second basis image and the one represented by the third basis image with $d = 7$), due to the face similarity. Also, notice that the subjects in the case of the third basis image are in shadow at the right part of the image.

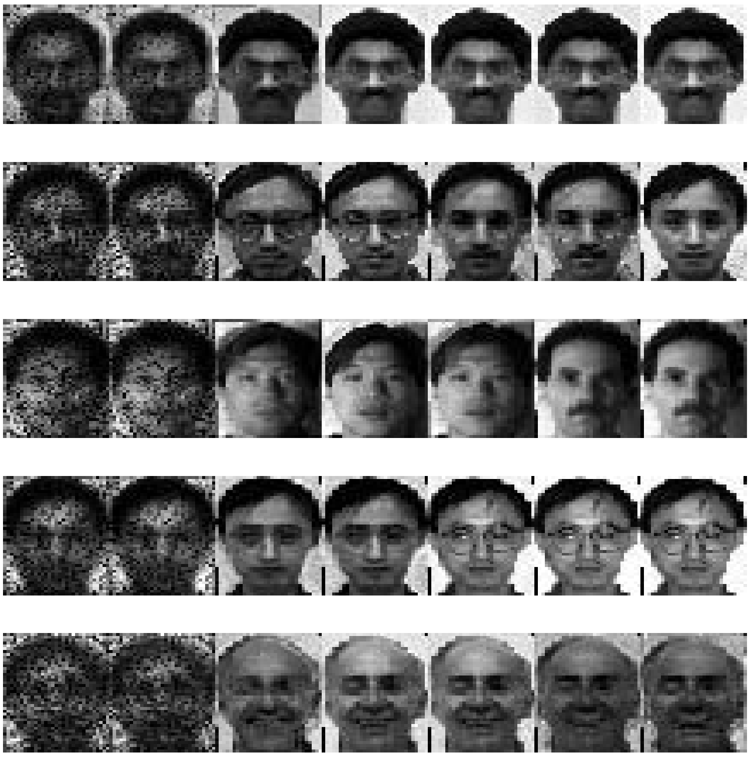

Fig. 3. Five different basis images retrieved by the PNMF with $d = \{2,3,4,5,6,7,8\}$ for the YALE database. While the basis images corresponding to $d = 2$ and 3 are quite noisy, as the degree increases more pixels are taken into account and the basis images retrieve person identity.

The results of face recognition for the YALE database are tabulated in Table II. The best performance is provided by PNMF for all three classifiers. KPCA and PCA performed equally for the CSM classifier with a slight improvement in accuracy of 0.2% for the RBFNN classifier. 100 % accuracy is yielded in the case of PNMF combined with RBFNN, while the same maximum accuracy is obtained by all methods when SVM classifier is used. Since the PNMF has a very good performance in this database where the subjects were recorded under various lighting conditions and different facial expressions, making the faces less recognizable, PNMF seems to be much more robust to illumination changes than the other tested approaches. This is emphasized by two aspects. First, by visual inspection of Figure 3, where basis images incorporate illumination effects in their representation, and, second, by comparing the results for all methods corresponding to the YALE database presented in Table II for the CSM classifier. For this classifier, which is the simplest of the three classifiers and thus, in this case, the superiority of a classifier-image representation pair can be mainly attributed to the image representation part of the pair, PNMF performed the best, yielding an accuracy 2 % higher than the second best technique, i.e., NMF.

The last database involved in the experiments was the AT&T. In this case the results are quite different from the ones obtained for the YALE database. The best results are obtained by the KICA followed by LNMF and PCA when combined with CSM. These results are consistent with the ones reported in [11] where the performance of LNMF is improved because this algorithm is rotation invariant (up to some degree), since it generates local features in contrast to NMF which yields more distributed features. Approximately the same performance is obtained by RBFNN, while SVM classifier performed much worse. In all cases, the PNMF method performs better than the algorithm that it tries to improve, namely the NMF algorithm, that yielded the second lowest performance. KPCA was the worst performing algorithm in this database.

As noted in the database description, AT&T database contains slightly rotated faces and we believe that this is the main reason that the PNMF algorithm does not perform well in this database. More specifically, the basis images generated by PNMF have a holistic (distributed) appearance, as is obvious in Figure 3 and thus the algorithm is vulnerable to rotation. This explanation is also supported by the following facts: a) the NMF algorithm that also generates holistic basis images performs poorly in the same database b) the LNMF algorithm that generates sparse (local) basis images (that are more robust to rotations) performs better than both NMF and PNMF in this database.

The approach we have developed for updating the basis images and the coefficients relies on iterative minimization. Obviously, other optimization techniques such as, for example, Sequential Quadratic Programming or the interior-reflective Newton method can be used. However, due to the fact that we deal here with a large-scale optimization (taking into account the vector dimension) these approaches can be prohibitive in terms of computational cost or memory requirements as was shown in the following experiment. Having evaluated the analytical expression for the derivative and the constraints for the cost function, we run the MATLAB [50] routine “fmincon” with the large-scale optimization option to tackle our problem and compared it with our iterative solution starting with the same initial random matrices **B** and **Z**. The routine “fmincon” for large-scale optimization uses the interior-reflective Newton method with the help of preconditioned conjugate gradients. The initial value of the cost function was found to be $Q_{initial} = 3.4610 \cdot 10^8$. Table III shows the final value Q_{final} of the cost function and the time necessary to reach the minimum for 9 basis images of 20 × 15 pixels. The methods provided slightly different values for the final cost function. This is due to the fact that both methods are only able to find local minima and they rely on different minimization procedure. The proposed algorithm which was also implemented in MATLAB, was executed almost 431 times faster than “fmincon”. We must also notice that we were not able to run “fmincon” with images having the dimension of 40 × 30 pixels and for more than 5 basis images due to the memory limitations.

V. Conclusions

In this paper we have modified the NMF algorithm in such manner that the discovered features have a non-linear dependency, while the decomposition factors remain non-negative. The underlying idea of the new factorization algorithm, named PNMF, is the usage of the polynomial kernel function, which causes the decomposition to take place in a feature space instead of the input space. The algorithm has been applied on two databases for the facial expression classification task and on two databases for face recognition. For comparison purposes six reference feature extraction algorithms (NMF, LNMF, PCA, KPCA, ICA and KICA) have been also used. The features retrieved by the aforementioned approaches have been classified by three classifiers CSM, NN, and SVM, respectively. Except for one database (AT&T) out of the four, PNMF outperforms the other approaches for all classifiers. The benefit of the proposed approach is evident in problems where the classes are difficult to separate, as in the case of the JAFFE database.

When comparing PNMF with the other two kernel-based approaches that have been proposed, namely KPCA and KICA, one can state that the major difference of PNMF with its counterparts is the non-negative nature of the nonlinear features that it retrieves. It turns out that, generally, these features possess powerful discriminant features. However experiments showed that there is no guarantee that PNMF is superior to KPCA or KICA for all data sets, a statement that is also applicable to all decomposition methods, e.g whcn comparing PCA against ICA (a comparison that resulted in contradicting results, depending on the data set). A situation where PNMF appears to be superior to KPCA (and to all other approaches, as implied by its good performance in the case of the YALE database) is in cases where illumination variations exist in the data, a case that KPCA is known to be unable to handle well. However, additional experiments (object recognition task) should be performed to strengthen our claim, or to verify it. We plan to conduct such experiments in the future. However, one can state with confidence that

PNMF is always better than its linear counterpart, i.e. the NMF algorithm, in terms of retrieving more powerful latent variables for pattern classification, as evidenced by the experimental results.

The way the development of the iterative approach for updating the decomposition factors was carried out in this paper, does not permit a non-negative decomposition for another kernel type, such as, for instance, the RBF kernel. This is due to the negative solution resulting from the derivative associated to the RBF kernel. Other approaches that allow a more flexible kernel have to be found, which can be a topic of future work. Using other kernel types could be a potential way to improve the performance of the kernel non-negative matrix factorization approach

VI. Aknowledgement

This work has been conducted in conjunction with the "SIMILAR" European Network of Excellence on Multimodal Interfaces of the IST Programme of the European Union (www.similar.cc).

Appendix

A. Derivation of the polynomial KNMF coefficients update

For updating the expression of the polynomial KNMF coefficients we present two approaches which lead to the same updating rule. The first approach derives the multiplicative rule (7) based on finding an upper bound minimizer which iteratively moves towards tighter upper bounds of the cost function involved. The second approach utilizes a gradient descent optimization procedure.

Definition 2 The function $G(b, b^{(t)})$ is an upper bound for $Q(b)$ if, for any b and $b^{(t)}$ we have $G(b, b) = Q(b)$ and $G(b, b^{(t)}) \geq Q(b)$, $\forall b \neq b^{(t)}$ [28].

Lemma 1 If G is an upper bound for Q, then Q is decreasing under the update $b^{(t+1)} = \text{argmin}_b G(b, b^{(t)})$.

Proof: $Q(b^{(t+1)}) = G(b^{(t+1)}, b^{(t+1)}) \leq G(b^{(t+1)}, b^{(t)}) \leq G(b^{(t)}, b^{(t)}) = Q(b^{(t)})$ ■

Lemma 2 Let δ_{ij} denote the Kronecker delta function and let $\mathbf{L}$ be a diagonal matrix with elements $L_{ij} = \delta_{ij}(\mathbf{K}_{zz}\mathbf{b})_i/b_i^{(t)}$. Then the following theorem holds:

Theorem .1: The upper bound of the function

$$Q(b) = \frac{1}{2}\sum_{j=1}^{n}\left(\Phi(\mathbf{x}_q) - \sum_{r=1}^{p} b_r\Phi(\mathbf{z}_r)\right)^2 \tag{A-1}$$

is given by:

$$G(b, b^{(t)}) = G(b^{(t)}) + (b - b^{(t)})^T\nabla Q(b^{(t)}) + \frac{1}{2}(b - b^{(t)})^T L(b^{(t)})(b - b^{(t)}), \tag{A-2}$$

where $\nabla Q(b^{(t)}) = \frac{\partial Q(b^{(t)})}{\partial b^{(t)}}$ is the first partial derivative with respect to $b^{(t)}$.

Proof: The cost function $Q(b)$ can be written as Taylor expansion in the neighborhood of the fixed point $b^{(t)}$ as follows:

$$Q(b) = Q(b^{(t)}) + (b - b^{(t)})^T\nabla Q(b - b^{(t)}) + \frac{1}{2}(b - b^{(t)})^T\nabla^2 Q(b^{(t)})(b - b^{(t)}), \tag{A-3}$$

where $\nabla^2 Q(b^{(t)}) = \frac{\partial^2 Q(b^{(t)})}{\partial b'^2}$ is the second partial derivative with respect to $b^{(t)}$. Obviously when $b = b^{(t)}$ we have $G(b, b) = Q(b)$. For $b \neq b^{(t)}$, $G(b, b^{(t)}) \geq Q(b)$ is explicitly given by:

$$(b - b^{(t)})^T(L(b^{(t)}) - K_{zz})(b - b^{(t)}) \geq 0, \tag{A-4}$$

taking into account that $\frac{\partial^2 Q(b^{(t)})}{\partial b'^2} = \mathbf{K}_{zz}$. The relation (A-4) is equivalent with the statement that the matrix $\mathbf{L} - \mathbf{K}_{zz}$ is positive semidefinite. In order to prove that, consider first the matrix $\mathbf{P}$ whose elements are of the form:

$$P_{ij} = b_i^{(t)}(L - K_{zz})_{ij}b_j^{(t)}. \tag{A-5}$$

The matrix $\mathbf{P}$ is generated by rescaling elementwise the elements of $\mathbf{L} - \mathbf{K}_{zz}$. Therefore, $\mathbf{L} - \mathbf{K}_{zz}$ is positive semidefinite if $\mathbf{P}$ is positive semidefinite. For $\mathbf{P}$ and for any $\mathbf{b}$ we have:

$$\begin{aligned}
\mathbf{b}^T\mathbf{P}\mathbf{b} &= \sum_{i,j} b_i P_{ij} b_j \qquad (A\text{-}6)\\
&= \sum_{i,j} b_i b_j b_j^t \delta_{ij} (\mathbf{K}_{zz}\mathbf{b})_i - \sum_{i,j} b_i^t b_j^t b_i b_j K_{ij}^{zz}\\
&= \sum_{i,j} b_i^t b_j^t K_{ij}^{zz} b_i^2 - \sum_{i,j} b_i^t b_j^t b_i b_j K_{ij}^{zz}\\
&= \frac{1}{2}\sum_{i,j} b_i^t b_j^t b_i^2 K_{ij}^{zz} + \frac{1}{2}\sum_{i,j} b_i^t b_j^t K_{ij}^{zz} b_j^2 - \sum_{i,j} b_i^t b_j^t b_i b_j K_{ij}^{zz}\\
&= \frac{1}{2}\sum_{i,j} K_{ij}^{zz} b_i^t b_j^t (b_i - b_j)^2 \geq 0.
\end{aligned}$$

■

Here, K_{ij}^{zz} is the $\{i,j\}$ element of the matrix $\mathbf{K_{zz}}$.

Derivation of eq. (7), first solution.

Proof: Since $G(b, b^{(t)})$ is un upper bound for $Q(b)$ and $b^{(t+1)} = \mathrm{argmin}_b G(b, b^{(t)})$ we find its minimum by taking the derivative and setting it to zero:

$$\frac{\partial G(b, b^{(t)})}{\partial b} = \nabla Q(b^{(t)}) + L(b^{(t)})(b - b^{(t)}) = 0. \qquad (A\text{-}7)$$

This gives us:

$$L(b^{(t)})b = L(b^{(t)})b^{(t)} - \nabla Q(b^{(t)}). \qquad (A\text{-}8)$$

Multiplying on the left by $L(b^{'})^{-1}$, we get:

$$b = b^{(t)} - L(b^{(t)})^{-1} \nabla Q(b^{(t)}). \qquad (A\text{-}9)$$

The partial derivative of $\nabla Q(b^{(t)})$ with respect to $b^{(t)}$ is given by:

$$\begin{aligned}
\frac{\partial Q(b)}{\partial b_q} = -\Phi(\mathbf{z}_q) \sum_{j=1}^{n} \left(\Phi(\mathbf{x}_q) - \sum_{r=1}^{p} b_r \Phi(\mathbf{z}_r) \right) = \\
-\left(\sum_{j=1}^{n} \Phi(\mathbf{z}_q)\Phi(\mathbf{x}_q) - \sum_{j=1}^{n}\sum_{r=1}^{p} b_r \Phi(\mathbf{z}_q)\Phi(\mathbf{z}_r) \right) = \qquad (A\text{-}10)\\
= -(\mathbf{k}_{zx} - \mathbf{K}_{zz}\mathbf{b})
\end{aligned}$$

Since $\mathbf{L}(b^{(t)})$ is a diagonal matrix,

$$L_{ij}(b^{(t)})^{-1} = b_i^{(t)} \frac{1}{\delta_{ij}(\mathbf{K}_{zz}\mathbf{b})_i}. \qquad (A\text{-}11)$$

By substituting (A-11) and (A-10) in (A-9), we obtain

$$\begin{aligned} b_i &= b_i^{(t)} + b_i^{(t)} \frac{1}{(\mathbf{K}_{zz}\mathbf{b})_i}((\mathbf{k}_{zx})_i - (\mathbf{K}_{zz}\mathbf{b})_i) \\ &= b_i^{(t)} + b_i^{(t)} \frac{(\mathbf{k}_{zx})_i}{(\mathbf{K}_{zz}\mathbf{b})_i} - b_i^{(t)} \frac{(\mathbf{K}_{zz}\mathbf{b})_i}{(\mathbf{K}_{zz}\mathbf{b})_i} \\ &= b_i^{(t)} \frac{(\mathbf{k}_{zx})_i}{(\mathbf{K}_{zz}\mathbf{b})_i}. \end{aligned} \tag{A-12}$$

Putting it in a matrix form, we obtain the expression (7). ■

Derivation of eq. (7), second solution.

Proof: An alternative solution can be found if we use a gradient descent optimization such as:

$$b = b^{(t)} - \eta(\nabla Q(b^{(t)})), \tag{A-13}$$

with $0 < \eta < \frac{1}{\beta}$,where η is the learning step and $\beta > 0$. Taking the Taylor expansion (A-3) and substituting b from (A-13), we finally have:

$$Q(b) - Q(b^{(t)}) = \eta(\nabla^2 Q(b^{(t)}))\left(1 - \frac{1}{2}\beta\eta\right). \tag{A-14}$$

Choosing an appropriate value for η and α such as $\eta = L_{ij}$ and $\beta = K_{zz}$, we have $\eta < \frac{1}{\beta}$, therefore $\left(1 - \frac{1}{2}\beta\eta\right) > 0$ for any element $z \in [0, 1]$, hence $Q(b) > Q(b^{(t)})$. However, this approach leads to the same solution since the relation (A-13) is equivalent with (A-9) after substituting η and β. ■

B. Derivation of the polynomial KNMF basis images update, i.e. of eq. (8)

Proof: The same rationale is followed for obtaining an update rule for the basis images by employing eq. (11). Taking all images, the partial derivative of $\nabla Q(z)$ with respect to z is given by:

$$\frac{\partial Q(z)}{\partial z_{\mu i}} = -\sum_{j=1}^{n} b_\mu \mathbf{K}^{'}(\mathbf{x}_j \cdot \mathbf{z}_\mu) x_{ji} + \sum_{r=1}^{p} b_r b_\mu \mathbf{K}^{'}(\mathbf{z}_r \cdot \mathbf{z}_\mu) z_{ri}. \tag{A-15}$$

In this case, the relation $G(z, z^{(t)}) \geq Q(z)$ translates into the following:

$$\frac{1}{2}\sum_{ij}[dzbK_{zz}^{d-1} - d(d-1)z^2bK_{zz}^{d-2} + d(d-1)x^2K_{xz}^{d-2}]z_i^{(t)}z_j^{(t)}(z_i - z_j)^2 \geq 0, \tag{A-16}$$

which is equivalent with:

$$x^2 K_{xz}^{d-2} \geq z^2 b K_{zz}^{d-2}. \tag{A-17}$$

Finally, the following inequality holds:

$$x^2 K_{xz}^{d-2} \geq x^2 K_{zz}^{d-2} \geq z^2 b K_{zz}^{d-2}, \tag{A-18}$$

since $(\mathbf{x}^T\mathbf{z})^{d-2} \geq (\mathbf{z}^T\mathbf{z})^{d-2}$, $\forall x \in [0, 255], z \in [0, 1]$ and $d \geq 2$, with equality for $d = 2$. Further, by choosing $L_{ij} = \delta_{ij}(\mathbf{z}\boldsymbol{\omega}\mathbf{K}_{zz})_i / z_i^{(t)}$ we come up with the updating expression for basis images in (8). ■

References

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

[2] P. O. Hoyer, "Modeling receptive fields with non-negative sparse coding," *Neurocomputing*, vol. 52-54, pp. 547–552, 2003.

[3] D. Guillamet, B. Schiele, and J. Vitri, "Analyzing non-negative matrix factorization for image classification," in *Proc. of 16th Int. Conf. on Pattern Recognition*, vol. II, pp. 116–119, 2002.

[4] P. Paatero and U. Tapper, "Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, pp. 111–126, 1994.

[5] T. Kawamoto, K. Hotta, T. Mishima, J. Fujiki, M. Tanaka, and T. Kurita "Estimation of single tones from chord counds using non-negative matrix factorization," in *Neural Network World*, vol. 3, pp. 429–436, 2000.

[6] B. Wang and M. D. Plumbley, "Musical audio stream separation by non-negative matrix factorization," in *Proc. of DMRN Summer Conference*, Glasgow, 2005.

[7] M. Cooper and J. Foote, "Summarizing video using non-negative similarity matrix factorization," in *Proc. IEEE Workshop on Multimedia Signal Processing*, pp. 25–28, 2002.

[8] E. Kim, P. K. Hopke, and E. S. Edgerton, "Source identification of Atlanta aerosol by positive matrix factorization," *Journal Air Waste Manage. Assoc.*, vol. 53, no. 1, pp. 731–739, 1977.

[9] J. Piper, V. P. Pauca, R. J. Plemmons, and M. Giffin, "Unmixing spectral data for space objects using independent component analysis and non-negative matrix factorization," in *Proc. Amos Technical Conf.*, 2004.

[10] P. Pauca, F. Shahnaz, M. Berry and R. Plemmons, "Text mining using non-negative matrix factorization," in *Proc. SIAM Inter. Conf. on Data Mining*, 2004.

[11] S. Z. Li, X. W. Hou and H. J. Zhang, "Learning spatially localized, parts-based representation," *Int. Conf. Computer Vision and Pattern Recognition*, pp. 207–212, 2001.

[12] I. Buciu, N. Nikolaidis, and I. Pitas, “A comparative study of NMF, DNMF, and LNMF algorithms applied for face recognition,” in *Second IEEE-EURASIP International Symposium on Control, Communications, and Signal Processing*, 2006.

[13] I. Buciu and I. Pitas, “Application of non-negative and local non-negative matrix factorization to facial expression recognition,” *Int. Conf. on Pattern Recognition*, pp. 228–291, 2004.

[14] J. Eggert and E. Korner, “Sparse coding and NMF,” in *IEEE Int. Joint Conf. on Neural Networks*, pp. 2529–2533, 2004.

[15] O. Schwartz and E. P. Simoncelli, “Natural signal statistics and sensory gain control,” *Nature Neuroscience*, vol. 4, no. 8, pp. 819–825, 2001.

[16] E. Simoncelli, “Vision and the statistics of the visual environment,” *Current Opinion in Neurobiology*, vol. 13, pp. 144–149, 2003.

[17] J. Touryan, G. Felsen, and Y. Dan, “Spatial structure of complex cell receptive fields measured with natural images,” *Neuron*, vol. 45, pp. 781–791, 2005.

[18] J. Rapela, J. M. Mendel, and N. M. Grzywacz, “Estimating nonlinear receptive fields from natural images,” *Journal of Vision*, vol. 6, no. 4, pp. 441–474, 2006.

[19] J. Malo, E. P. Simoncelli, I. Epifanio, and R. Navarro, “Non-linear image representation for efficient perceptual coding,” *IEEE Trans. on Image Processing*, vol. 15, no. 1, pp. 68–80, 2006.

[20] M. S. Bartlett, H. M. Lades and T. K. Sejnowski, “Independent component representations for face recognition,” *Proc. SPIE Conf. Human Vision and Electronic Imaging III*, vol 3299, pp. 528–539, 1998.

[21] K. R. Müller, S. Mika, G. Rätsch, K. Tsuda and B. Schölkopf, “An introduction to kernel-based learning lagorithms,” *IEEE Trans. Neural Networks*, vol. 12, no. 2, pp. 181–201, 2001.

[22] A. S. Have, M. A. Girolami and J. Larsen, “Clustering via kernel decomposition,” *IEEE Trans. on Neural Networks*, vol. 17, no. 1, pp. 48–58, 2006.

[23] F. R. Bach and M. J. Jordan, “Kernel independent component analysis,” *Machine Learning Research*, vol. 3, pp. 1–48, 2002.

[24] C. S. Ong, A. J. Smola, and R. C. Williamson, “Learning the kernel with hyperkernels,” *Journal of Machine Learning Research*, vol. 6, pp. 1043–1071, 2005.

[25] I. Wai-Hung Tsang and J. Tin-Yau Kwok, “Efficient hyperkernel learning using second-order cone programming,” *IEEE Trans. on Neural Networks*, vol. 17, no. 1, pp. 48–58, 2006.

[26] S. Yang, S. Yan, C. Zhangand X. Tang, “Bilinear analysis for kernel selection and nonlinear feature extraction,” *IEEE Trans. on Neural Networks*, vol. 18, no. 5, pp. 1442–1452, 2007.

[27] R. Basri and D. W. Jacobs, “Lambertian reflectance and linear subspaces,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 218–233, 2003.

[28] D D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” *Advances Neural Information Processing Systems*, vol. 13, pp. 556–562, 2001.

[29] S. Kullback and R. Leibler, “On information and sufficiency,” *Annals of Mathematical Statistics*, no. 22, pp.

79–86, 1951.

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

[31] L. M. 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, no. 7, pp. 200–217, 1967.

[32] I. S. Dhillon and S. Sra, "Generalized Nonnegative Matrix Approximations with Bregman Divergences," in *Neural Information Processing Systems*, vol. 18, 2005.

[33] J. S. - Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, 2004.

[34] C. J. C. Burges, "Simplified support vector decision rules," in *Int. Conf. on Machine Learning*, pp. 71–77, 1996.

[35] B. Schölkopf, S. Mika, C. J. C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A. J. Smola, "Input space versus feature space in kernel-based methods," *IEEE Trans. on Neural Networks*, vol. 10, no. 5, pp. 1000–1017, 1999.

[36] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, John Wiley & Sons, Toronto, 2001.

[37] I. T. Jolliffe, *Principal Component Analysis*, (2nd ed.), New York: Springer-Verlag, 2002.

[38] F. R. Bach and M. J. Jordan, "Kernel independent component analysis," *Machine Learning Research*, vol. 3, pp. 1–48, 2002.

[39] B. Schölkopf, A. J. Smola, and K.- R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[40] T. Kanade, J. Cohn and Y. Tian, "Comprehensive database for facial expression analysis," *in Proc. IEEE Inter. Conf. on Face and Gesture Recognition*, pp. 46–53, 2000.

[41] M. Pantic and L. J. M. Rothkrantz, "Expert system for automatic analysis of facial expressions," *Image and Vision Computing*, vol. 18, no. 11, pp. 881–905, 2000.

[42] M. J. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding Facial Expressions with Gabor Wavelets," in *Proc. Third IEEE Int. Conf. Automatic Face and Gesture Recognition*, pp. 200–205, 1998.

[43] http://cvc.yale.edu

[44] http://www.uk.research.att.com/

[45] V. Vapnik, *The nature of statistical learning theory*, Springer-Verlag, 1995.

[46] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," *Advances in Kernel Methods - Support Vector Learning*, vol. 12, pp. 185–208, 1999.

[47] J. C. Platt, N. Cristianini, and J. S.-Taylor, "Large margin DAGs for mutliclass classification," *Advances in Neural Information Procesing Systems*, vol. 12, pp. 547–553, 2000.

[48] R. Ma and J. Wang, "Automatic facial expression recognition using linear and nonlinear holistic spatial analysis," in *First Int. Conf. on Affective Computing and Intelligent Interaction*, pp. 144–151, 2005.

[49] G. Donato, M. S. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski, “Classifying facial actions,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 974–989, 1999.

[50] http://www.mathworks.com/

TABLE I

Maximum accuracy (%) obtained for the various methods used in the facial expression classification experiments. The minimum number of basis images corresponding to the maximum accuracy is also presented. The degree of the polynomial PNMF is given in parenthesis. The best result is shown in bold.

Database	Classifier	Maximum accuracy (%)/number of basis images						
		NMF	LNMF	PNMF	ICA	PCA	KICA	KPCA
C-K	CSM	77.4/36	81.4/25	**81.8**/16 ($d = 6$)	71.4/25	72.9/16	74.3/36	72.9/64
	RBFNN	67.1/81	77.1/25	**78.6**/81 ($d = 2$)	72.9/16	74.3/25	72.9/16	74.3/100
	SVM	78.6/100	81.4/81	**83.9**/100 ($d = 2$)	80/64	81.4/100	82.9/25	82.9/100
JAFFE	CSM	66.3/81	62.4/49	**69.8**/25 ($d = 5$)	63.4/100	61.3/100	66.7/49	58/16
	RBFNN	61.9/81	60.3/25	**65**/16 ($d = 4$)	61.9/121	60.3/49	61.9/64	55/25
	SVM	74.6/49	74.6/36	**77.8**/49 ($d = 6$)	74.6/36	74.6/36	76.2/81	71.4/81
JAFFE_{diff}	CSM	70/121	89.3/49	**93.8**/100 ($d = 7$)	91/16	90.1/16	89.3/25	91/49
	RBFNN	76.8/25	82.1/25	**87.5**/100 ($d = 7$)	82.1/25	85.7/25	82.1/36	86/25
	SVM	78.6/36	82.1/25	**83.9**/25 ($d = 5$)	82.1/16	82.1/16	82.1/49	82.5/64

TABLE II

Maximum accuracy (%) obtained for the various methods used in the face recognition experiments. The minimum number of basis images corresponding to the maximum accuracy is also presented. The degree of the polynomial PNMF is given in parenthesis. The best result is shown in bold.

Database	Classifier	Maximum accuracy (%)/number of basis images						
		NMF	LNMF	PNMF	ICA	PCA	KICA	KPCA
YALE	CSM	93.2/49	87.8/64	**95.2**/25 ($d = 7$)	90.6/49	88/25	93.2/49	88/81
	RBFNN	98.3/64	98.3/64	**100**/9 ($d = 2$)	98.1/25	99.3/9	98/49	99.5/100
	SVM	100/9	100/9	100/9 ($d = 2$)	100/9	100/9	100/9	100/9
AT&T	CSM	94.1/49	96/100	94.1/64 ($d = 3$)	93.5/81	95/100	**98**/64	86.5/121
	RBFNN	90/36	**95.5**/100	92.5/100 ($d = 6$)	93/49	94.5/196	94.5/49	85/49
	SVM	70.5/25	72.5/36	71.5/81 ($d = 7$)	**73**/49	**73**/49	**73**/49	70/64

TABLE III

Convergence time (in seconds), initial and final value for the cost function Q for the iterative (PNMF) and "fmincon" methods, respectively. The number of basis images is 9 and the dimension of the basis image is 20×15 pixels.

Method	Time (*seconds*)	$Q_{initial}$	Q_{final}
PNMF	50	$3.4610 \cdot 10^8$	$2.3077 \cdot 10^4$
fmincon	21548	$3.4610 \cdot 10^8$	$2.2270 \cdot 10^4$