Nonlinear Nonnegative Component Analysis

Stefanos Zafeiriou and Maria Petrou
Dept. of Electrical and Electronic Engineering
Imperial College London
South Kensington Campus SW7 2AZ
{s.zafeiriou,maria.petrou}@imperial.ac.uk

Abstract

In this paper general solutions for Nonlinear Nonnegative Component Analysis for data representation and recognition are proposed. That is, motivated by a combination of the Nonnegative Matrix Factorization (NMF) algorithm and kernel theory, which has lead to an NMF algorithm in a polynomial feature space [1], we propose a general framework where one can build a nonlinear nonnegative component analysis using kernels, the so-called Projected Gradient Kernel Nonnegative Matrix Factorization (PGKNMF). In the proposed approach, arbitrary positive kernels can be adopted while at the same time it is ensured that the limit point of the procedure is a stationary point of the optimization problem. Moreover, we propose fixed point algorithms for the special case of Radial Basis Function (RBF) kernels. We demonstrate the power of the proposed methods in face and facial expression recognition applications.

1. Introduction

In computer vision and pattern recognition fields, one of the most popular ways to represent an object is by writing it as a linear combination of basis. The basis is in many cases used to extract features and/or find a low dimensionality object representation to be subsequently used for recognition. One of the most popular methods to find a basis is *Principal Component Analysis* (PCA) [2]. Another very popular method, that works on the statistical independence of the basis objects or the weights of the representation, is the Independent Component Analysis (ICA) [3]. ICA has been widely used for the problem of face recognition. In this paper, we deal with the problem of object representation using images and we are particularly interested in face recognition problems.

In [4] a decomposition of objects using a linear basis was proposed by considering non-negativity constraints for both the basis and the weights of the linear combination, the so-called Nonnegative Matrix Factorization (NMF). NMF, like PCA, represents an image as a linear combination of basis images. NMF does not allow negative elements in either the basis images or the representation coefficients used in the linear combination of the basis images. Thus, it represents an image only by additions of weighted basis images. The non-negativity of constraints arises in many real image processing applications, since the pixels in a grayscale image have non-negative intensities. As claimed in [4], an object (represented as an image) is more naturally coded into its parts by using only additions between the different bases.

Both NMF and PCA are linear models, thus they may fail to model efficiently the nonlinearities that are present in most real life applications. Nonlinear component analysis is a research topic that has been greatly developed in the past decade [5, 6, 1]. This is mainly attributed to the great success of combining Support Vector Machines (SVMs) with kernels [5]. Since then, kernels have been widely used for finding non-linear counterparts of PCA, the so-called Kernel Principal Component Analysis (KPCA) [5] and for descovering nonlinear high order dependencies of data, the so-called Kernel Independent Component Analysis (KICA) [6]. Recently, a nonlinear counterpart of NMF has been proposed, the so-called Polynomial Nonnegative Matrix Factorization (PNMF) [1]. The PNMF has been partly motivated by biological issues like yielding a model compatible with the neurophysiology paradigms (non-negativity constraints and nonlinear image decomposition [7]) and has been used to discover higher-order correlations between image pixels that may lead to more powerful latent features. For more details on the motivation of PNMF the interested reader may refer to [1]. The main drawback of [1] is that only polynomial kernels can be used. In this paper we propose methods in which one can use, apart from polynomial kernels other popular positive kernels.

Finally, we should comment that in [8] the original NMF method was applied to the kernel matrix of the original data. This is different from the approach proposed in [1] and the approach followed in this paper. In our case the problem is

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

formulated so as to find a set of nonnegative weights and nonnegative vectors such that the projected training vectors can be written as a linear combination of the learned projected vectors, under the nonlinear mapping. On the other hand, in [8] the aim was to find a nonnegative decomposition of the kernel matrix which was just the application of NMF to a nonnegative matrix of inner products.

In this paper we propose a general method for nonlinear nonnegative component analysis using arbitrary positive kernels in order to remedy the above mentioned limitations of PNMF [1] and the NMF of kernel matrices [8]. Moreover, we present a method for nonlinear nonnegative component analysis using RBF kernels.

2. Problem Formulation

In this paper we consider the problem of representing facial images in a nonlinear way. Every facial image is scanned row-wise to form an image vector $\mathbf{x}_i \in \Re_+^F$. Assume that we have a database of M images in total. The problem of PNMF, in [1], has been formulated as following. Let $\phi : \Re_+^F \to \mathcal{H}$ be a mapping that projects image $\mathbf{x}_i$ to a Hilbert space $\mathcal{H}$ of arbitrary dimensionality. Our aim is to find a set of vectors $\mathbf{z}_j \in \Re_+^F$ and a set of weights $h_{j,i} \geq 0$ such as:

$$\phi(\mathbf{x}_i) \approx \sum_j h_{j,i}\phi(\mathbf{z}_j), \tag{1}$$

or more generally:

$$\mathbf{X}^\Phi \approx \mathbf{Z}^\Phi \mathbf{H} \tag{2}$$

where $\mathbf{X}^\Phi = [\phi(\mathbf{x}_1)\dots\phi(\mathbf{x}_M)]$, $\mathbf{Z}^\Phi = [\phi(\mathbf{z}_1)\dots\phi(\mathbf{z}_P)]$ and $[\mathbf{H}]_{j,i} = h_{j,i}$ with $\mathbf{H} \in \Re_+^{P\times M}$. Vectors $\mathbf{z}_j$ are the so-called pre-images [5, 9] of the approximation. The dot product in $\mathcal{H}$ is written by means of kernels as $k(\mathbf{x}_i, \mathbf{x}_j) = \langle\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)\rangle = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$.

In order to find the preimage matrix $\mathbf{Z} = [\mathbf{z}_1\dots\mathbf{z}_P]$ and the weights matrix $\mathbf{H}$, the least squares error is used for measuring the error of the approximation:

$$D_\phi(\mathbf{X}^\Phi, \mathbf{Z}^\Phi\mathbf{H}) = \frac{1}{2}\sum_{i=1}^{M} ||\phi(\mathbf{x}_i) - \sum_j h_{j,i}\phi(\mathbf{z}_j)||^2. \tag{3}$$

The optimization problem is as follows:

$$\min_{z_{i,k}\geq 0,\ h_{k,j}\geq 0} D_\phi(\mathbf{X}^\Phi, \mathbf{Z}^\Phi\mathbf{H}). \tag{4}$$

In order to provide further motivation for the approach we may express problem (4) as follows. Given a database of images $\mathbf{X} \in \Re_+^{F\times M}$ and a nonlinear mapping ϕ we want to find a matrix $\mathbf{Z}$ of preimages $\mathbf{z}_i$ of the same domain as $\mathbf{x}_i$ (i.e., if $\mathbf{x}_i$ are nonnegative grayscale images then we want the preimages $\mathbf{z}_j$ to be nonnegative images, as well). After the projection, under mapping ϕ, we want the weights $h_{j,i}$ of the linear combination to be nonnegative. This is motivated by the biological aspect that the firing rates of the neural visual system are nonnegative, which has also motivated NMF and PNMF [4, 1]. Moreover, by not allowing negative values for $h_{j,i}$ we have $\phi(\mathbf{x}_i) \approx \sum_j h_{j,i}\phi(\mathbf{z}_j)$, thus $k(\mathbf{x}_i, \cdot) \approx \sum_j h_{j,i}k(\mathbf{z}_j, \cdot)$ which is always positive (for positive kernels $k(\mathbf{x}_i, \cdot)$).

In [1], in order to solve the constrained optimization problem (4) the authors used auxiliary functions for both $\mathbf{H}$ and $\mathbf{Z}$. Before proceeding in describing how the algorithm in [1] has been formulated, we should define the following matrices:

$$\begin{array}{l} [\mathbf{K}_{x,x}]_{i,j} = \langle\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)\rangle = k(\mathbf{x}_i, \mathbf{x}_j) \\ [\mathbf{K}_{z,z}]_{i,j} = \langle\phi(\mathbf{z}_i), \phi(\mathbf{z}_j)\rangle = k(\mathbf{z}_i, \mathbf{z}_j) \\ [\mathbf{K}_{z,x}]_{i,j} = \langle\phi(\mathbf{z}_i), \phi(\mathbf{x}_j)\rangle = k(\mathbf{z}_i, \mathbf{x}_j) \\ \mathbf{K}_{x,z} = \mathbf{K}_{z,x}^T. \end{array} \tag{5}$$

In [1], the kernel considered was the polynomial kernel:

$$k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T\mathbf{x}_j)^d \tag{6}$$

where d was the degree of the polynomial. Assuming that vectors $\mathbf{x}_i$ are linearly independent, then matrix $\mathbf{K}_{x,x}$ (which has the properties of a Gram matrix [5]) is positive definite. The same holds for the Gram matrix $\mathbf{K}_{z,z}$, in the case that $\mathbf{z}_j$ are linearly independent. In all cases, both $\mathbf{K}_{x,x}$ and $\mathbf{K}_{z,z}$ are at least positive semidefinite matrices.

In order to calculate the solution of the optimization problem, Buciu et al [1] defined auxiliary functions and minimized them in order to obtain a set of updating rules. Although they defined nice multiplicative updating rules, these rules hold only for polynomial kernels. In the following we shall propose a procedure where a wide variety of kernels can be used in the decomposition and the limit point of the procedure is guaranteed to be a stationary point of the optimization problem.

Before describing the proposed algorithms, we shall briefly describe the difference between the Nonlinear Nonnegative Component Analysis proposed in this paper and the Nonnegative Matrix Factorization on Kernels in [8].

In our approach we consider the problem of approximating $\mathbf{X}^\Phi$ using a matrix $\mathbf{Z}^\Phi$ and a matrix of weights $\mathbf{H}$ with $\mathbf{x}_i$ and $\mathbf{z}_j$ being in the same domain (i.e., $\Re_+^F$) (this problem is formally written in (2)). On the other hand in [8] they considered an easier and more restricted problem, namely the problem of finding a nonnegative decomposition of matrix $\mathbf{X}^{\Phi^T}\mathbf{X}^\Phi = \mathbf{K}_{x,x}$. That is, they setup the problem of approximating $\mathbf{K}_{x,x} \in \Re_+^{M\times M}$:

$$\mathbf{K}_{x,x} \approx \mathbf{G}\mathbf{W} \tag{7}$$

with two nonnegative matrices $\mathbf{G} \in \Re_+^{M\times P}$ and $\mathbf{W} \in \Re_+^{P\times M}$ using the NMF algorithm in [4]. As it can be seen, the above problem is just the application of NMF to matrix $\mathbf{K}_{x,x}$.

In the followed approach, we not only find an approximation of the kernel matrix $\mathbf{K}_{x,x}$ but we also identify the pre-images $\mathbf{Z}^{\Phi}$, as well. The latter cannot be achieved by following the above procedure.

3. Projected Gradient Methods for Nonlinear Non-Negative Matrix Factorization

Using the notion of kernels, metric (3), that quantifies the approximation of the vectors in $\mathbf{X}^{\Phi}$ as a linear combination of the basis in $\mathbf{Z}^{\Phi}$, can be expanded as:

$$\begin{aligned} D_{\phi} \quad & (\mathbf{X}^{\Phi}, \mathbf{Z}^{\Phi}\mathbf{H}) = \tfrac{1}{2}\textstyle\sum_{i=1}^{M} ||\phi(\mathbf{x}_i) - \sum_{j=1}^{P} h_{j,i}\phi(\mathbf{z}_j)||^2 \\ & = \tfrac{1}{2}\textstyle\sum_{i=1}^{M}(k(\mathbf{x}_i, \mathbf{x}_i) - 2\sum_{j=1}^{P} h_{j,i}k(\mathbf{z}_j, \mathbf{x}_i) \\ & + \textstyle\sum_{j=1}^{P}\sum_{l=1}^{P} h_{j,i}h_{l,i}k(\mathbf{z}_l, \mathbf{z}_j)) \end{aligned} \tag{8}$$

The minimization of (8) subject to nonnegative constraints for the weights matrix $\mathbf{H}$ and the basis matrix $\mathbf{Z}$ yields the nonlinear nonnegative decomposition. This optimization problem will be solved using projected gradients in order to guarantee that the limit point is stationary and that the nonnegativity constraints of $\mathbf{z}_i$ and $\mathbf{h}_j$ (the j-th column of $\mathbf{H}$) are met. In order to find the limit point, two functions are defined:

$$f_{\mathbf{Z}}(\mathbf{H}) = D_{\phi}(\mathbf{X}^{\Phi}, \mathbf{Z}^{\Phi}\mathbf{H}) \text{ and } f_{\mathbf{H}}(\mathbf{Z}) = D_{\phi}(\mathbf{X}^{\Phi}, \mathbf{Z}^{\Phi}\mathbf{H}) \tag{9}$$

by keeping $\mathbf{Z}$ and $\mathbf{H}$ fixed, respectively.

The projected gradient method used in this paper, successively optimizes two subproblems [10]:

$$\begin{aligned} & \min_{\mathbf{Z}} f_{\mathbf{H}}(\mathbf{Z}) \\ & \text{subject to } z_{i,k} \geq 0, \end{aligned} \tag{10}$$

and

$$\begin{aligned} & \min_{\mathbf{H}} f_{\mathbf{Z}}(\mathbf{H}) \\ & \text{subject to } h_{k,j} \geq 0. \end{aligned} \tag{11}$$

The first partial derivative with respect to $h_{a,b}$ is:

$$\frac{\partial f_{\mathbf{Z}}}{\partial h_{a,b}} = [(\mathbf{Z}^{\Phi^T}\mathbf{Z}^{\Phi})\mathbf{H} - \mathbf{Z}^{\Phi^T}\mathbf{X}^{\Phi}]_{a,b} = [\mathbf{K}_{z,z}\mathbf{H} - \mathbf{K}_{z,x}]_{a,b}. \tag{12}$$

For the first partial derivative with respect to $z_{a,b}$ we have:

$$\begin{aligned} \tfrac{\partial f_{\mathbf{Z}}}{\partial z_{a,b}} \quad & = \textstyle\sum_{i=1}^{M}(-h_{b,i}\frac{\partial k(\mathbf{z}_b,\mathbf{x}_i)}{\partial z_{a,b}} + \\ & \tfrac{1}{2}(\textstyle\sum_{l=1}^{P} h_{b,i}h_{l,i}\frac{\partial k(\mathbf{z}_l,\mathbf{z}_b)}{\partial z_{a,b}} + \\ & \textstyle\sum_{l=1,l\neq b}^{P} h_{b,i}h_{l,i}\frac{\partial k(\mathbf{z}_l,\mathbf{z}_b)}{\partial z_{a,b}})). \end{aligned} \tag{13}$$

The projected gradient KNMF method is an iterative method that comprises two main phases. These two phases are iteratively repeated until the ending condition is met or the number of iterations exceeds a given number. In the first phase, an iterative procedure is followed for the optimization of (10), while in the second phase, a similar procedure is followed for the optimization of (11). At the beginning, the basis matrix $\mathbf{Z}^{(1)}$ and the weight matrix $\mathbf{H}^{(1)}$ are initialized randomly, in such a way that their entries are nonnegative.

The procedure followed for the minimization of the two subproblems is iteratively (similar to the one used in [10]) followed until the global convergence rule is met:

$$\begin{aligned} & ||\nabla f(\mathbf{H}^{(t)})||_F + ||\nabla f(\mathbf{Z}^{(t)})||_F \leq \\ & \epsilon\left(||\nabla f(\mathbf{H}^{(1)})||_F + ||\nabla f(\mathbf{Z}^{(1)})||_F\right) \end{aligned} \tag{14}$$

which checks the stationarity of the solution pair $(\mathbf{H}^{(t)}, \mathbf{Z}^{(t)})$.

Matrix $\mathbf{Z}$ may be subsequently used for extracting features as follows. Let $\mathbf{y}$ be a vector such that $\mathbf{y} \in \Re_{+}^{F}$. Then, the projected vector $\tilde{\mathbf{y}} \in \Re^{P}$ is calculated as follows:

$$\tilde{\mathbf{y}} = \mathbf{Z}^{\Phi^\dagger}(\phi(\mathbf{y}) - \mathbf{m}^{\Phi}) \tag{15}$$

where $\mathbf{m}^{\Phi} = \frac{1}{M}\sum_i \phi(\mathbf{x}_i)$, the $\mathbf{Z}^{\Phi^\dagger}$ is the pseudo-inverse of $\mathbf{Z}^{\Phi}$ and is calculated as:

$$\mathbf{Z}^{\Phi^\dagger} = (\mathbf{Z}^{\Phi^T}\mathbf{Z}^{\Phi})^{-1}\mathbf{Z}^{\phi^T} = \mathbf{K}_{z,z}^{-1}\mathbf{Z}^{\phi^T}. \tag{16}$$

The inverse $\mathbf{K}_{z,z}^{-1}$ can be calculated, in most cases, and since usually $P \ll M$, thus $\mathbf{K}_{z,z}$ is full ranked.

Now, using (16) feature extraction (15) may be reformulated as:

$$\tilde{\mathbf{y}} = \mathbf{K}_{z,z}^{-1}\mathbf{Z}^{\Phi^T}(\phi(\mathbf{y}) - \mathbf{m}^{\Phi}) = \mathbf{K}_{z,z}^{-1}\mathbf{g}(\mathbf{y}) \tag{17}$$

where $\mathbf{g}(\mathbf{y}) = [k(\mathbf{z}_1, \mathbf{y}) - \frac{1}{M}\sum_i k(\mathbf{x}_i, \mathbf{y}), \ldots, k(\mathbf{z}_P, \mathbf{y}) - \frac{1}{M}\sum_i k(\mathbf{x}_i, \mathbf{y})]^T$. In [1] $\mathbf{Z}$ was directly used for feature extraction as $\tilde{\mathbf{y}} = \mathbf{Z}^{\dagger}\mathbf{y}$. This procedure leads to only linear feature extraction.

4. A Nonlinear Nonnegative Component Analysis Approach for RBF Kernels

In this section we consider the problem of nonlinear nonnegative component analysis using RBF kernels and we propose an alternative fixed point algorithm for the minimization of the cost. In kernel methods the RBF kernel is given by:

$$k(\mathbf{x}_i, \mathbf{x}_j) = e^{-||\mathbf{x}_i - \mathbf{x}_j||^2/s} \tag{18}$$

where s is the spread of the Gaussian function.

Let us consider the problem as follows. First we consider the kernel expansion:

$$\mathbf{g}_i = \sum_j h_{j,i}\phi(\mathbf{z}_j) \tag{19}$$

and now we seek to approximate it by $\acute{\mathbf{g}}_i = \gamma_i\phi(\mathbf{x}_i)$. We allow $\gamma_i \neq 1$, which is reasonable, since the length of $\mathbf{g}_i$ is not crucial for building decision functions [5]. The problem is to minimize the following:

$$f(\mathbf{Z}, \mathbf{H}) = \sum_i ||\mathbf{g}_i - \acute{\mathbf{g}}_i||^2 = \sum_i || \sum_j h_{j,i}\phi(\mathbf{z}_j) - \gamma_i \phi(\mathbf{x}_i)||^2. \tag{20}$$

As in the algorithm in the previous section we consider solving the two partial minimization problems:

$$\min_{h_{k,j} \geq 0} f(\mathbf{h}_j) \tag{21}$$

and

$$\min_{z_{i,k} \geq 0} f(\mathbf{z}_k) \tag{22}$$

in an iterative manner, where $f(\mathbf{h}_j)$ and $f(\mathbf{z}_k)$ are equal to $f(\mathbf{Z}, \mathbf{H})$ by keeping all but $\mathbf{h}_j$ and $\mathbf{z}_k$ constant, respectively. For the minimization problem in (21) we consider the actual optimization problem (20), while for the minimization (22) we consider a transformed version of the problem.

In the t-th iteration for the solution of subproblem (21) we follow the procedure of the auxiliary function. That is, we identify a solution via the definition and the optimization of a proper auxiliary function. In our case we choose an auxiliary function similar to the one used in [1]. The updating rules are the following:

$$h_{j,i}^{(t)} = h_{j,i}^{(t-1)} \frac{\gamma_i [\mathbf{K}_{x,z}^{(t-1)}]_{j,i}}{[\mathbf{K}_{z,z}^{(t-1)} \mathbf{h}_i^{(t-1)}]_j} \tag{23}$$

or in matrix notation

$$\mathbf{H}^{(t)} \leftarrow \mathbf{H}^{(t-1)} \otimes \mathbf{K}_{x,z}^{(t-1)} \oslash (\mathbf{K}_{z,z}^{(t-1)} \mathbf{H}^{(t-1)}) \otimes \mathbf{C} \tag{24}$$

with $[\mathbf{K}_{x,z}]_{i,j} = -e^{||\mathbf{x}_i - \mathbf{z}_j||^2/s}$, $[\mathbf{K}_{z,z}]_{i,j} = -e^{||\mathbf{z}_i - \mathbf{z}_j||^2/s}$ and $\mathbf{C}_{j,i} = \gamma_i$. Operator $\otimes$ is used for denoting element-wise matrix multiplication while $\oslash$ denotes element-wise matrix division.

We use a similar reasoning as the one followed in [5] for finding the preimages of kernel algorithms, in order to specify the updating rules in subproblem (22). That is, as in [5] we minimize the orthogonal projection of $\phi(\mathbf{x}_i)$ onto $\sum_j h_{j,i}\phi(\mathbf{z}_j)$:

$$\min_{\mathbf{z}_j, \gamma_i} \sum_{i=1}^{N} || \frac{\langle \phi(\mathbf{x}_i), \sum_j h_{j,i}\phi(\mathbf{z}_j) \rangle}{\langle \sum_j h_{j,i}\phi(\mathbf{z}_j), \sum_j h_{j,i}\phi(\mathbf{z}_j) \rangle} - \phi(\mathbf{x}_i)|| \tag{25}$$

which is equivalent to:

$$\max_{\mathbf{z}_j, \gamma_i} d_{RBF}(\mathbf{Z}) = \sum_{i=1}^{N} \frac{\langle \phi(\mathbf{x}_i), \sum_j h_{j,i}\phi(\mathbf{z}_j) \rangle^2}{\langle \sum_j h_{j,i}\phi(\mathbf{z}_j), \sum_j h_{j,i}\phi(\mathbf{z}_j) \rangle}. \tag{26}$$

In order to simplify further optimization problem (26) optimize only:

$$\max_{\mathbf{z}_j, \gamma_i} \sum_{i=1}^{N} \langle \phi(\mathbf{x}_i), \sum_j h_{j,i}\phi(\mathbf{z}_j) \rangle. \tag{27}$$

By using fixed point iteration algorithms like [5] (i.e., setting $\frac{\partial d(\mathbf{Z})}{\partial z_{i,k}} = 0$), update rules can be derived for $z_{i,k}$ as:

$$z_{i,k}^{(t)} = \frac{\sum_j x_{i,j} h_{k,j} k(\mathbf{z}_k^{(t-1)}, \mathbf{x}_j)}{\sum_j h_{k,j} k(\mathbf{z}_k^{(t-1)}, \mathbf{x}_j)}. \tag{28}$$

In compact matrix notation the update rules may be written as:

$$\mathbf{Z}^{(t)} \leftarrow \mathbf{X}(\mathbf{H}^{(t)} \otimes \mathbf{K}_{z,x}^{(t-1)}) \oslash (\mathbf{B}^{(t-1)}) \tag{29}$$

where $\mathbf{B}^{(t-1)}$ has as rows the diagonal of matrix $\mathbf{H}^{(t)}\mathbf{K}_{x,z}^{(t-1)}$. After the iterations in (28), the optimal $\gamma_i^{(t)}$ is given by:

$$\gamma_i^{(t)} = \langle \phi(\mathbf{x}_i), \sum_j h_{j,i}^{(t)} \phi(\mathbf{z}_j^{(t)}) \rangle = \sum_j h_{j,i}^{(t)} k(\mathbf{x}_i, \mathbf{z}_j^{(t)}). \tag{30}$$

The algorithm proposed in this Section is referred to as KNMF-RBF in the rest of the paper.

4.1. Another Formulation

Instead of finding both pre-images $\mathbf{z}_j$ and $\mathbf{H}$ simultaneously we follow a different strategy. First we assume that every $\phi(\mathbf{z}_j)$ can be written as linear combination of $\phi(\mathbf{x}_i)$ i.e,:

$$\phi(\mathbf{z}_j) \quad = \textstyle\sum_{i=1}^{N} m_{i,j}\phi(\mathbf{x}_i). \tag{31}$$

The corresponding optimization problem is given by:

$$\min_{m_{ik} \geq 0, h_{ki} \geq 0} D_\phi(\mathbf{X}^\Phi, \mathbf{X}^\Phi \mathbf{M}\mathbf{H}) = ||\mathbf{X}^\Phi - \mathbf{X}^\Phi \mathbf{M}\mathbf{H}||^2. \tag{32}$$

It can be proven that the update rules:

$$\begin{aligned} \mathbf{M}^{(t)} \quad &= \mathbf{M}^{(t-1)} \otimes \Big((\mathbf{K}_{x,x} \mathbf{H}^{(t-1)^T}) \\ &\oslash (\mathbf{K}_{x,x} \mathbf{M}^{(t-1)^{(t)}} \mathbf{H}^{(t-1)} \mathbf{H}^{(t-1)^T}) \Big)^p \\ \mathbf{H}^{(t)} \quad &= \mathbf{H}^{(t-1)} \otimes \Big((\mathbf{M}^{(t)^T} \mathbf{K}_{x,x}) \\ &\oslash (\mathbf{M}^{(t)^T} \mathbf{K}_{x,x} \mathbf{M}^{(t)} \mathbf{H}^{(t-1)}) \Big)^p \end{aligned} \tag{33}$$

where $p = 1$ or $p = \frac{1}{2}$, guarantee that the function (32) remains nonincreasing.

After, the convergence of the above sequence the pre-images $\mathbf{z}_j$ can now be found by the following optimization problem:

$$\begin{aligned} &\min_{\mathbf{z}_j} ||\phi(\mathbf{z}_j) - \beta_j \textstyle\sum_{i=1}^{N} m_{i,j}\phi(\mathbf{x}_i)||^2 \\ &\text{subject to } z_{kj} \geq 0. \end{aligned} \tag{34}$$

The above optimization can be solved using many algorithms [5]. One of them is the projected gradient algorithms presented in 3. For the special case of RBF functions we can use following update rules for obtaining $\mathbf{z}_j$ with $j = 1, \ldots, P$:

$$\mathbf{z}_j^{(t+1)} = \frac{\sum_{i=1}^{N} m_{ij} k(\mathbf{z}_j^{(t)}, \mathbf{x}_i)\mathbf{x}_i}{\sum_{i=1}^{N} m_{ij} k(\mathbf{z}_j^{(t)}, \mathbf{x}_i)}. \tag{35}$$

5. Experimental Results

5.1. Facial Expressions Recognition Experiments with the Jaffe Database

This database used for experiments contains 213 images of Japanese female facial expressions (JAFFE) [11]. Ten subjects produced 3 or 4 examples of each of the six facial expressions plus a neutral pose, thus producing a total of 213 images of facial expressions. The difference images, created by subtracting the neutral image intensity values from the corresponding values of the facial expression image, were calculated. Each difference image was initially normalized, resulting in an image built only from positive values. For the experimental procedure, 150 difference images were used for training and the remaining 63 were used for testing. As it can be seen from Table 1, the proposed method achieved the best facial expression recognition results.

5.2. Face Verification Experiments using the XM2VTS database

The experiments conducted with the XM2VTS database using the protocol described in [12]. The images were aligned semi-automatically according to the eyes position of each facial image using the eye coordinates. The facial images were down-scaled to a resolution of 64×64 pixels. Histogram equalization was used for the normalization of the facial image luminance.

The XM2VTS database contains 295 subjects, 4 recording sessions and two shots (repetitions) per recording session. It provides two experimental setups, namely, Configuration I and Configuration II [12]. Each configuration is divided into three different sets: the training set, the evaluation set and the test set. The training set is used to create client and impostor models for each person. The evaluation set is used to learn the verification decision thresholds. In case of multimodal systems, the evaluation set is also used to train the fusion manager [12]. For both configurations the training set has 200 clients, 25 evaluation impostors and 70 test impostors. The two configurations differ in the distribution of client training and client evaluation data. For additional details concerning the XM2VTS database an interested reader can refer to [12].

The experimental procedure followed in the experiments was the one also used in [12]. For comparison reasons the same methodology using Configuration I of the XM2VTS database was used. The performance of the algorithms is quoted by the Equal Error Rate (EER) which is the scalar figure of merit that is often used to judge the performance of a verification algorithm. The comparisons of the best EER achieved in the XM2VTS database can be found in Figure 2.

5.3. Face Recognition Using Photometric Stereo

In this Section we describe experiments of face recognition using photometric stereo. We collected a database of faces by setting a device for proper capture of images. The four intensity images were processed using a standard photometric stereo method [13, 14, 15]. This results in a dense field of surface normals, which we then integrate to form height maps. The albedo and the depth images were manually aligned according to the eye coordinates and scaled to resolution 90×100.

The device was installed in the offices of General Dynamics. Staff and visitors were kindly asked to use it. After a period of more than 6 months, more than 250 persons had used it. For 113 persons there are images that had been taken more than a week interval. For the majority of them (about 90), there are samples with more than one month apart. For the experiments presented here we have a very challenging experimental procedure using only one grayscale albedo image for training and one grayscale albedo image for testing. Moreover, one depth image is used for training and one for testing.

As we have already mentioned, most of training and testing images had been taken with more than one month interval and most of the training and testing images display a different facial expression. The recognition rate versus the dimensionality for the grayscale albedo is plotted in Figure 1a, while the recognition rate for the depth images is plotted in Figure 1b. As it can be seen, the proposed approaches achieved the best recognition rates (i.e, PGKNMF for albedo images and KNMF-RBF for depth images), as well.

6. Conclusions

In this paper, we proposed a method for nonlinear nonnegative matrix factorization using projected gradients. Unlike other methods for this purpose, the proposed method allows the use of a wide variety of kernels, apart from the polynomial kernels considered in [1]. Moreover, it guarantees that the limit point of the algorithms is a stationary point of the optimization procedure. For the special case of RBF kernels, a fixed point algorithm is proposed for nonnegative nonlinear component analysis. The experimental results have shown that the proposed methods can be successfully used for feature extraction and recognition and lead to better classification rates when compared with well-known and widely used nonlinear feature extraction techniques (like KPCA and KICA).

Acknowledgment

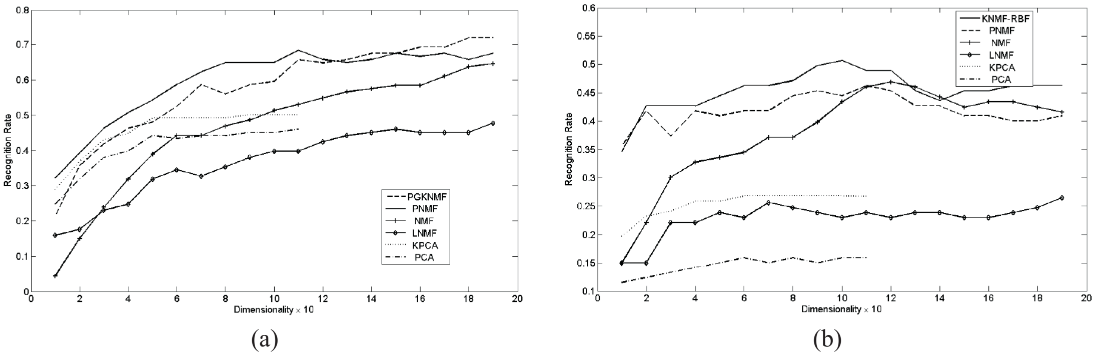
This work has been supported by the EPSRC project EP/E028659/1 Face Recognition using Photometric Stereo.

Table 1. Best accuracy (%)/number of basis images, for the difference images of the Jaffe database

NMF	LNMF	PNMF	ICA	PCA	KICA	KPCA	PGKNMF
70/121	89.3/49	93.8/100	91/16	90.1/16	89.3/25	91/49	**95**/100

Table 2. Best EER (%) for the XM2VTS database

	NMF	LNMF	PNMF	ICA	PCA	KICA	KPCA	PGKNMF
EER%	8.5	8.2	5.4	4.1	4.3	3.5	3.4	3.4



(a)

(b)

Figure 1. a) Recognition rates for the grayscale albedo images; b) Recognition rates for the depth images.

References

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

[2] K. Fukunaga, *Statistical Pattern Recognition*, CA: Academic, San Diego, 1990.

[3] A Hyvarinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, Wiley Interscience, 2001.

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

[5] B. Scholkopf and A. Smola, *Learning with Kernels*, MIT Press, Cambridge, MA, 2002.

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

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

[8] D. Zhang, Z.-H. Zhou, and S. Chen, "Non-negative matrix factorization on kernels," in *Proceedings of the 9th Pacific Rim International Conference on Artificial Intelligence (PRICAI'06)*, 2006, vol. LNAI 4099.

[9] J.T.-Y. Kwok and I.W.-H. Tsang, "The pre-image problem in kernel methods," *IEEE Transactions on Neural Networks*, vol. 15, no. 6, pp. 1517–1525, 2004.

[10] C.-J. Lin, "Projected gradients for nonnegative matrix factorization," *Neural Computation accepted for publication*, 2007.

[11] M. J. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with gabor wavelets," in *in Proc. Third IEEE Int Conf. Automatic Face and Gesture Recognition*, 1998, pp. 200–205.

[12] K. Messer and al. et, "XM2VTSDB: The extended M2VTS database," in *AVBPA'99*, Washington, DC, USA, 22-23 March 1999, pp. 72–77.

[13] S. Barsky and M. Petrou, "The 4-source photometric stereo technique for three-dimensional surfaces in the presence of highlights and shadows," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, , no. 25, pp. 1239–1252, 2003.

[14] S. Barsky and M. Petrou, "Design issues for a colour photometric stereo system," *J. Math. Imaging Vis.*, , no. 24, pp. 143–162, 2006.

[15] V. Argyriou and M. Petrou, "Photometric stereo: An overview," *Advances in Imaging and Electronic Physics*, vol. 156, pp. 1–55, 2008.