Zhirong Yang and Jorma Laaksonen. 2007. Multiplicative updates for non-negative projections. Neurocomputing, volume 71, numbers 1-3, pages 363-373.

Available online at www.sciencedirect.com
ScienceDirect
Neurocomputing 71 (2007) 363–373
NEUROCOMPUTING
www.elsevier.com/locate/neucom

# Multiplicative updates for non-negative projections

Zhirong Yang*, Jorma Laaksonen

*Laboratory of Computer and Information Science, Helsinki University of Technology, P.O. Box 5400, FI-02015 HUT, Espoo, Finland*

Received 18 April 2006; received in revised form 20 September 2006; accepted 9 November 2006
Communicated by S. Choi
Available online 20 February 2007

**Abstract**

We present here how to construct multiplicative update rules for non-negative projections based on Oja's iterative learning rule. Our method integrates the multiplicative normalization factor into the original additive update rule as an additional term which generally has a roughly opposite direction. As a consequence, the modified additive learning rule can easily be converted to its multiplicative version, which maintains the non-negativity after each iteration. The derivation of our approach provides a sound interpretation of learning non-negative projection matrices based on iterative multiplicative updates—a kind of Hebbian learning with normalization. A convergence analysis is scratched by interpretating the multiplicative updates as a special case of natural gradient learning. We also demonstrate two application examples of the proposed technique, a non-negative variant of the linear Hebbian networks and a non-negative Fisher discriminant analysis, including its kernel extension. The resulting example algorithms demonstrate interesting properties for data analysis tasks in experiments performed on facial images.


*Keywords:* Non-negative projection; Multiplicative update; Oja's rule; Hebbian learning; Matrix factorization

## 1. Introduction

Projecting high-dimensional input data into a lower-dimensional subspace is a fundamental research topic in signal processing and pattern recognition. Non-negative projection is desired in many real-world applications, for example, for images, spectra, etc., where the original data are non-negative. However, most classical subspace approaches such as *principal component analysis* (PCA) and *Fisher discriminant analysis* (FDA), which are solved by *singular value decomposition* (SVD), fail to produce the non-negativity property.

Recently, Lee and Seung [12,13] introduced iterative multiplicative updates, which are based on the decomposition of the gradient of given objective function, for non-negative optimizations. They applied the technique to the *non-negative matrix factorization* (NMF) which seems to yield sparse representations. Several variants of NMF such as [5,22,24] have later been proposed, where the original NMF objective function is combined with various regularization terms. More recently, Yuan and Oja [23] presented a method called *projective non-negative matrix factorization* (P-NMF) without any additional terms, but directly derived from the objective function of PCA networks except that the projection was constrained to be non-negative. The simulation results of P-NMF indicate that it can learn highly localized and non-overlapped part-based basis vectors. However, none of the above works provides an explanation why the multiplicative updates can produce sparser and more localized base components.

The multiplicative update rules of the above algorithms are based on decomposition of the gradients of an objective function into positive and negative parts, one as the numerator and the other as the denominator. Nevertheless, such method would fail when the gradient is not naturally expressed in positive and negative parts. Sha et al. [21] proposed an alternative decomposition of the gradient and applied it to the minimization of a quadratic objective.

*Corresponding author.
*E-mail addresses:* zhirong.yang@hut.fi (Z. Yang), jorma.laaksonen@hut.fi (J. Laaksonen).

0925-2312/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2006.11.023

This method albeit cannot handle the situation where the gradient contains only one positive (negative) term. Furthermore, how to combine orthogonality or quadratic unit norm constraints with this method is still unknown.

In this paper we present a more general technique to reformulate a variety of existing additive learning algorithms to their multiplicative versions in order to produce non-negative projections. The derivation is based on Oja's rule [17] which integrates the normalization factor into the additive update rule. Therefore, our method provides a natural way to form the numerator and denominator in the multiplicative update rule even if external knowledge of gradient decomposition is not available. Another major contribution of our approach is that its derivation also provides a sound interpretation of the non-negative learning based on iterative multiplicative updates—a kind of Hebbian learning with normalization.

We demonstrate applicability of the proposed method for two classical learning algorithms, PCA and FDA, as examples. In the unsupervised PCA learning, our multiplicative implementation of linear Hebbian networks outperforms the NMF in localized feature extraction, and its derivation provides an interpretation why P-NMF can learn non-overlapped and localized basis vectors. In the supervised FDA learning, our non-negative variant of the *linear discriminant analysis* (LDA) can serve as a feature selector and its kernel extension can reveal an underlying factor in the data and be used as a sample selector. The resulting algorithms of the above examples are verified by experiments on facial image analysis with favorable results.

The remaining of the paper is organized as follows. First we introduce the basic idea of multiplicative update rules in Section 2. The non-negative projection problem is then formulated in Section 3. In Section 4 we review Oja's rule and present the technique how to use it in forming the multiplicative update rules. The proposed method is applied in two examples in Section 5: one for unsupervised learning and the other for supervised. The experimental results of the resulting algorithms are presented in Section 6. Finally, conclusions are drawn in Section 7.

## 2. Multiplicative updates

Suppose there is an algorithm which seeks an $m$-dimensional solution $\mathbf{w}$ that maximizes an objective function $\mathscr{J}(\mathbf{w})$. The conventional *additive update* rule for such a problem is

$$\tilde{\mathbf{w}} = \mathbf{w} + \gamma \mathbf{g}(\mathbf{w}), \tag{1}$$

where $\tilde{\mathbf{w}}$ is the new value of $\mathbf{w}$, $\gamma$ a positive learning rate and the function $\mathbf{g}(\mathbf{w})$ outputs an $m$-dimensional vector which represents the *learning direction*, obtained e.g. from the gradient of the objective function. For notational brevity, we only discuss the learning for vectors in this section, but it is easy to generalize the results to the matrix case, where we will use capital letters $\mathbf{W}$ and $\mathbf{G}$.

The multiplicative update technique first generalizes the common learning rate to different ones for individual dimensions:

$$\tilde{\mathbf{w}} = \mathbf{w} + \mathrm{diag}(\boldsymbol{\eta})\mathbf{g}(\mathbf{w}), \tag{2}$$

where $\boldsymbol{\eta}$ is an $m$-dimensional positive vector. Choosing different learning rates for individual dimensions changes the update direction and hence this method differs from the conventional steepest-gradient approaches in the full real-valued domain.

It has been shown that the following choice of $\boldsymbol{\eta}$ has particular interesting properties in the constraint of non-negativity (see e.g. [12,21]). Suppose $\mathbf{w}$ is non-negatively initialized. If there exists a separation of the learning direction into two positive terms $\mathbf{g}(\mathbf{w}) = \mathbf{g}^+ - \mathbf{g}^-$ by some external knowledge, then one can choose $\eta_i = w_i/g_i^-$, $i = 1, \ldots, m$, such that the components of (2) become

$$\tilde{w}_i = w_i + \eta_i[\mathbf{g}(\mathbf{w})]_i = w_i + \frac{w_i}{g_i^-}(g_i^+ - g_i^-) = w_i \frac{g_i^+}{g_i^-}. \tag{3}$$

The above *multiplicative update* maintains the non-negativity of $\mathbf{w}$. In addition, $w_i$ increases when $g_i^+ > g_i^-$, i.e. $[\mathbf{g}(\mathbf{w})]_i > 0$, and decreases if $[\mathbf{g}(\mathbf{w})]_i < 0$. Thus the multiplicative change of $w_i$ indicates how much the direction of that axis conforms to the learning direction. There exists two kinds of stationary points in the iterative use of the multiplicative update rule (3): one satisfies $g_i^+ = g_i^-$, i.e. $\mathbf{g}(\mathbf{w}) = \mathbf{0}$, which is the same condition for local optima as in the additive updates (1), and the other is $w_i \to 0$. The latter condition distinguishes the non-negative optimization from conventional ones and yields sparsity in $\mathbf{w}$, which is desired in many applications. Furthermore, unlike steepest gradient or exponential gradient [10], the multiplicative update rule (3) does not require any user-specified learning rates, which facilitates its application.

In most non-negative algorithms that use multiplicative updates (e.g. [13,21,24]), the convergence of the objective has been proven via an auxiliary function. However, such a function depends on particular update rules and sometimes could be difficult to find. Here we present a novel interpretation of multiplicative updates as an optimization using *natural gradient* [1], which greatly simplifies the convergence analysis of the objective. Define the matrix $\mathscr{G}(\mathbf{w})$ as

$$[\mathscr{G}(\mathbf{w})]_{ij} = \delta_{ij} \frac{g_i^-}{w_i}, \tag{4}$$

with $\delta_{ij}$ the Kronecker delta. The tensor $\mathscr{G}$ defines a Riemannian inner product

$$\langle d\mathbf{w}, d\mathbf{w} \rangle_{\mathbf{w}} = \sum_{i=1}^{m} \sum_{j=1}^{m} [\mathscr{G}(\mathbf{w})]_{ij} [d\mathbf{w}]_i [d\mathbf{w}]_j = \sum_{i=1}^{m} \frac{g_i^-}{w_i} [d\mathbf{w}]_i^2 \geqslant 0. \tag{5}$$

Since $\mathscr{G}(\mathbf{w})$ is diagonal, its inverse can be computed by

$$\left[(\mathscr{G}(\mathbf{w}))^{-1}\right]_{ij} = \delta_{ij} \frac{w_i}{g_i^-}. \tag{6}$$

Henceforth we can obtain the natural gradient ascend update [1] for $\mathbf{w}$:

$$\tilde{w}_i = w_i + \zeta \sum_{j=1}^{m} \left[(\mathscr{G}(\mathbf{w}))^{-1}\right]_{ij} g_i = w_i + \zeta \frac{w_i}{g_i^-}(g_i^+ - g_i^-), \qquad (7)$$

where $\zeta$ is a positive scalar. Setting $\zeta = 1$, we obtain the multiplicative update rule (3). Because the natural gradient is known to be the steepest direction in a Riemannian manifold [1], the multiplicative updates form a steepest gradient ascend method in $(0, +\infty)^m$ which is curved by the tangent of the given objective function. Therefore, the multiplicative update rule (3) guarantees monotonic increase of the objective function if $\zeta = 1$ corresponds to a sufficiently small learning step and thus (7) forms a good approximation of the continuous flow in the Riemannian space.

## 3. Non-negative projection

Subspace projection methods are widely used in signal processing and pattern recognition. An $r$-dimensional subspace out of $\Re^m$ can be represented by an $m \times r$ orthogonal matrix $\mathbf{W}$. In many applications one can write the objective function for selecting the projection matrix in the form

$$\underset{\mathbf{W}}{\text{maximize}}\ \mathscr{J}(\mathbf{W}) = \tfrac{1}{2}E\{F(\|\mathbf{W}^{\mathrm{T}}\mathbf{v}\|^2)\}, \qquad (8)$$

where $\mathbf{v}$ is an input vector, $F$ a function from $\Re$ to $\Re$, and $E\{\cdot\}$ denotes the expectation. For problems where $F(x) = x$, objective (8) can be simplified to

$$\underset{\mathbf{W}}{\text{maximize}}\ \mathscr{J}(\mathbf{W}) = \tfrac{1}{2}\mathrm{Tr}(\mathbf{W}^{\mathrm{T}}E\{\mathbf{v}\mathbf{v}^{\mathrm{T}}\}\mathbf{W}). \qquad (9)$$

Such form covers the objectives of many classical analysis methods such as PCA and Fisher's LDA. The motivation and neural architecture of (9) is justified in [11]. By setting $\mathbf{A} = E\{\mathbf{v}\mathbf{v}^{\mathrm{T}}\}$, we can write the gradient of (9) as

$$\frac{\partial \mathscr{J}(\mathbf{W})}{\partial \mathbf{W}} = E\{\mathbf{v}\mathbf{v}^{\mathrm{T}}\}\mathbf{W} = \mathbf{A}\mathbf{W}. \qquad (10)$$

Obviously $\mathbf{A}$ is a positive semi-definite matrix.

The function $F$ can be other than $F(x) = x$. For example, in [8] the log likelihood function $F(x) = \log p(x)$ was used and a variant of *independent component analysis* (ICA) was derived. In that case, $\mathbf{A} = E\{F'(\|\mathbf{W}^{\mathrm{T}}\mathbf{v}\|^2)\mathbf{v}\mathbf{v}^{\mathrm{T}}\}$ is a negative semi-definite matrix.

In summary, we consider a particular set of additive update rules with the learning direction $\mathbf{G}(\mathbf{W}) = \mathbf{A}\mathbf{W}$, where $\mathbf{A}$ is an $m \times m$ symmetric matrix. *Non-negative projection* requires that all elements of $\mathbf{W}$ are non-negative. For brevity, we only discuss the case where $\mathbf{A}$ is positive semi-definite. The derivation can easily be modified for the opposite case where $\mathbf{A}$ is negative semi-definite.

## 4. Oja's rule in learning non-negative projections

The multiplicative update rule described in Section 2 maintains the non-negativity. However, the gradient of projection objective yields a single term and does not provide any natural way to obtain $\mathbf{g}^+$ and $\mathbf{g}^-$ (or $\mathbf{G}^+$ and $\mathbf{G}^-$). In this section we present a very simple approach to include an additional term for constructing the multiplicative update rules if the solution is constrained to be of unit $L_2$-norm or orthogonal.

First let us look at the projection on a one-dimensional subspace. In many optimization problems the objective function $\mathscr{J}(\mathbf{w})$ is accompanied with the constraint

$$\omega(\mathbf{w}) = (\mathbf{w}^{\mathrm{T}}\mathbf{B}\mathbf{w})^{1/2} = 1, \qquad (11)$$

with $\mathbf{B}$ an $m \times m$ symmetric matrix. In particular, $\omega(\mathbf{w}) = \|\mathbf{w}\|$ if $\mathbf{B} = \mathbf{I}$. If $\mathbf{w}$ is initialized to fulfill (11), the normalization step

$$\mathbf{w}^{\mathrm{new}} = \tilde{\mathbf{w}}(\omega(\tilde{\mathbf{w}}))^{-1} \qquad (12)$$

maintains that the new $\mathbf{w}$ still satisfies (11) since $\omega(\beta\mathbf{w}) = \beta\omega(\mathbf{w})$ for a scalar $\beta$.

One may try to combine the two update steps (1) and (12) into a single step version. The normalization factor $(\omega(\mathbf{w}))^{-1}$ can be expressed using Taylor expansion as

$$\begin{aligned}(\omega(\tilde{\mathbf{w}}))^{-1} &= \left((\mathbf{w}+\gamma\mathbf{g})^{\mathrm{T}}\mathbf{B}(\mathbf{w}+\gamma\mathbf{g})\right)^{-1/2} \\ &= \left(1 + \gamma(\mathbf{w}^{\mathrm{T}}\mathbf{B}\mathbf{g} + \mathbf{g}^{\mathrm{T}}\mathbf{B}\mathbf{w}) + O(\gamma^2)\right)^{-1/2} \\ &\approx 1 - \tfrac{1}{2}\gamma(\mathbf{w}^{\mathrm{T}}\mathbf{B}\mathbf{g} + \mathbf{g}^{\mathrm{T}}\mathbf{B}\mathbf{w}). \end{aligned} \qquad (13)$$

Here $\mathbf{g} = \mathbf{g}(\mathbf{w})$ for brevity and the final step is obtained by dropping all terms of $O(\gamma^2)$ or higher orders. Inserting this result and (1) into (12), we obtain the following Oja's single-step update rule [17]:

$$\mathbf{w}^{\mathrm{new}} \approx \mathbf{w} + \tfrac{1}{2}\gamma(2\mathbf{g} - \mathbf{w}\mathbf{w}^{\mathrm{T}}\mathbf{B}\mathbf{g} - \mathbf{w}\mathbf{g}^{\mathrm{T}}\mathbf{B}\mathbf{w}), \qquad (14)$$

where again the terms of $O(\gamma^2)$ have been dropped.

Now setting $\mathbf{g}(\mathbf{w}) = \mathbf{A}\mathbf{w}$, we obtain a possible decomposition of $\mathbf{g}(\mathbf{w})$ into two non-negative parts as

$$\mathbf{g}(\mathbf{w}) = \mathbf{A}\mathbf{w} = \mathbf{A}^+\mathbf{w} - \mathbf{A}^-\mathbf{w}, \qquad (15)$$

where

$$A_{ij}^+ = \begin{cases} A_{ij} & \text{if } A_{ij} > 0 \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad A_{ij}^- = \begin{cases} -A_{ij} & \text{if } A_{ij} < 0, \\ 0 & \text{otherwise.} \end{cases} \qquad (16)$$

The simple update rule

$$w_i^{\mathrm{new}} = w_i \frac{[\mathbf{A}^+\mathbf{w}]_i}{[\mathbf{A}^-\mathbf{w}]_i} \qquad (17)$$

or [21]

$$w_i^{\mathrm{new}} = w_i \frac{[\mathbf{A}^+\mathbf{w}]_i}{\sqrt{[\mathbf{A}^+\mathbf{w}]_i[\mathbf{A}^-\mathbf{w}]_i}} \qquad (18)$$

usually yields poor results for non-negative projection problems. The situation is even more problematic when $\mathbf{A}$

is non-negative, i.e. $\mathbf{A}^- = \mathbf{0}$. As an alternative, we here propose to substitute $\mathbf{g}$ from (15) into (14) and then obtain

$$\mathbf{w}^{\text{new}} \approx \mathbf{w} + \tfrac{1}{2}\gamma(2\mathbf{A}^+\mathbf{w} - 2\mathbf{A}^-\mathbf{w} - \mathbf{w}\mathbf{w}^{\mathrm{T}}\mathbf{B}\mathbf{A}\mathbf{w} - \mathbf{w}\mathbf{w}^{\mathrm{T}}\mathbf{A}\mathbf{B}\mathbf{w}). \tag{19}$$

Suppose $\mathbf{w}$ is initialized with values in $(0, 1)$ and fulfills (11). All terms without their leading sign in the right side of (19) are then positive if $\mathbf{w}^{\mathrm{T}}(\mathbf{B}\mathbf{A} + \mathbf{A}\mathbf{B})\mathbf{w} > 0$ for all positive $\mathbf{w}$. This condition defines the convexity of the solution subspace and is satisfied when $\mathbf{B}\mathbf{A} + \mathbf{A}\mathbf{B}$ is positive definite. Verifying such positive definity can easily be done before the iterative learning.

We can then apply the generalization technique described in Section 2 and obtain the following multiplicative update rule:

$$w_i^{\text{new}} = w_i \frac{2[\mathbf{A}^+\mathbf{w}]_i}{2[\mathbf{A}^-\mathbf{w}]_i + [\mathbf{w}\mathbf{w}^{\mathrm{T}}(\mathbf{B}\mathbf{A} + \mathbf{A}\mathbf{B})\mathbf{w}]_i}. \tag{20}$$

Since $\mathbf{B}$ is symmetric, (20) can be further simplified to

$$w_i^{\text{new}} = w_i \frac{[\mathbf{A}^+\mathbf{w}]_i}{[\mathbf{A}^-\mathbf{w}]_i + [\mathbf{w}\mathbf{w}^{\mathrm{T}}\mathbf{B}\mathbf{A}\mathbf{w}]_i} \tag{21}$$

because in that case

$$\mathbf{w}^{\mathrm{T}}\mathbf{B}\mathbf{A}\mathbf{w} = (\mathbf{w}^{\mathrm{T}}\mathbf{B}\mathbf{A}\mathbf{w})^{\mathrm{T}} = \mathbf{w}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{w} = \mathbf{w}^{\mathrm{T}}\mathbf{A}\mathbf{B}\mathbf{w}. \tag{22}$$

The original Oja's rule for a single vector has been generalized to the matrix case [18]. It combines the following additive learning rule and normalization steps:

$$\tilde{\mathbf{W}} = \mathbf{W} + \gamma\mathbf{G}(\mathbf{W}), \tag{23}$$

$$\mathbf{W}^{\text{new}} = \tilde{\mathbf{W}}(\mathbf{\Omega}(\tilde{\mathbf{W}}))^{-1}, \tag{24}$$

where $\mathbf{W}$ and $\mathbf{G}$ are $m \times r$ matrices and

$$\mathbf{\Omega}(\mathbf{W}) = \mathbf{I} \tag{25}$$

is the optimization constraint in the matrix case. If $\mathbf{\Omega}(\mathbf{W}) = (\mathbf{W}^{\mathrm{T}}\mathbf{W})^{1/2}$, the normalization factor can be approximated by

$$(\mathbf{\Omega}(\tilde{\mathbf{W}}))^{-1} \approx \mathbf{I} - \tfrac{1}{2}\gamma(\mathbf{W}^{\mathrm{T}}\mathbf{G} + \mathbf{G}^{\mathrm{T}}\mathbf{W}), \tag{26}$$

with similar derivation as in (13). Inserting (26) and (23) into (24), we obtain

$$\mathbf{W}^{\text{new}} \approx \mathbf{W} + \tfrac{1}{2}\gamma(2\mathbf{G} - \mathbf{W}\mathbf{W}^{\mathrm{T}}\mathbf{G} - \mathbf{W}\mathbf{G}^{\mathrm{T}}\mathbf{W}). \tag{27}$$

By again applying the generalization on $\gamma$ and inserting $\mathbf{G} = \mathbf{A}\mathbf{W}$, we can get the following multiplicative update rule:

$$W_{ij}^{\text{new}} = W_{ij} \frac{[\mathbf{A}^+\mathbf{W}]_{ij} + [\mathbf{W}\mathbf{W}^{\mathrm{T}}\mathbf{A}^-\mathbf{W}]_{ij}}{[\mathbf{A}^-\mathbf{W}]_{ij} + [\mathbf{W}\mathbf{W}^{\mathrm{T}}\mathbf{A}^+\mathbf{W}]_{ij}}. \tag{28}$$

Our method is suitable for problems with the constraint of the form $\mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I}$ or $\mathbf{w}^{\mathrm{T}}\mathbf{B}\mathbf{w} = 1$, but generally it does not work for $\mathbf{W}^{\mathrm{T}}\mathbf{B}\mathbf{W} = \mathbf{I}$ if $\mathbf{B} \neq \mathbf{I}$. This is because such constraint of $\mathbf{B}$-uncorrelatedness is probably overwhelmed by the non-negative learning which tends to yield high orthogonality.

If one only considers the projection on the Stiefel manifold (i.e. $\mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I}$), a more straightforward derivation can be obtained by using the natural gradient. Given a learning direction $\mathbf{G} = \mathbf{A}\mathbf{W}$, the natural gradient ascend update is [16]

$$\mathbf{W}_{\text{nat}}^{\text{new}} = \mathbf{W} + \gamma(\mathbf{G} - \mathbf{W}\mathbf{G}^{\mathrm{T}}\mathbf{W}). \tag{29}$$

Substituting $\mathbf{A} = \mathbf{A}^+ - \mathbf{A}^-$ and applying the reforming technique on $\gamma$, we obtain the same multiplicative update rule (28). This is not surprising because Oja's rule and natural gradient are two essentially equivalent optimization methods on the Stiefel manifold except that the former is based on the ordinary Euclidean metric while the latter on a canonical Riemannian metric [4].

## 5. Examples

In this section, we apply the above reforming technique to two known projection methods, PCA and LDA. Before presenting the details, it should be emphasized that we are not aiming at producing new algorithms to replace the existing ones for reconstruction or classification. Instead, the main purpose of these examples is to demonstrate the applicability of the technique described in the previous section and to help readers get more insight in the reforming procedure.

### 5.1. Non-negative linear Hebbian networks

Using multiplicative updates for non-negative optimization stems from the NMF proposed by Lee and Seung [12]. Given an $m \times n$ non-negative input matrix $\mathbf{V}$ where columns are the input samples, NMF seeks two non-negative matrices $\mathbf{W}$ and $\mathbf{H}$ which maximizes the following objective:

$$\mathscr{J}_{\text{NMF}}(\mathbf{W}, \mathbf{H}) = \|\mathbf{V} - \mathbf{W}\mathbf{H}\|_{\mathrm{F}}. \tag{30}$$

Here $\|\cdot\|_{\mathrm{F}}$ is the Frobenius matrix norm, defined as

$$\|\mathbf{Q}\|_{\mathrm{F}} = \sum_{ij} Q_{ij}^2 \tag{31}$$

for a matrix $\mathbf{Q}$. The authors of [12] derived the following multiplicative update rules of NMF:

$$W_{ij}^{\text{new}} = W_{ij} \frac{[\mathbf{V}\mathbf{H}^{\mathrm{T}}]_{ij}}{[\mathbf{W}\mathbf{H}\mathbf{H}^{\mathrm{T}}]_{ij}}, \tag{32}$$

$$H_{ij}^{\text{new}} = H_{ij} \frac{[\mathbf{W}^{\mathrm{T}}\mathbf{V}]_{ij}}{[\mathbf{W}^{\mathrm{T}}\mathbf{W}\mathbf{H}]_{ij}}. \tag{33}$$

NMF is not as good as PCA in minimizing the reconstruction error, but it was reported that NMF is able to extract some localized and part-based representations of the input samples [12]. To improve such localization effect, Yuan and Oja have recently developed a variant of NMF called P-NMF [23], which is derived from the following

optimization problem:

$$\underset{\mathbf{W}\geqslant 0}{\text{minimize}}\ \mathscr{J}_{\text{P-NMF}}(\mathbf{W}) = \|\mathbf{V} - \mathbf{W}\mathbf{W}^{\mathrm{T}}\mathbf{V}\|_{\mathrm{F}}. \tag{34}$$

That is, P-NMF replaces the matrix $\mathbf{H}$ with $\mathbf{W}^{\mathrm{T}}\mathbf{V}$ in the objective function. This change makes P-NMF also a variant of PCA whose objective is same as that of P-NMF except the non-negative constraint. The unconstrained gradient of $\mathscr{J}_{\text{P-NMF}}(\mathbf{W})$ is given by

$$\frac{\partial \mathscr{J}_{\text{P-NMF}}(\mathbf{W})}{\partial W_{ij}} = -2[\mathbf{V}\mathbf{V}^{\mathrm{T}}\mathbf{W}]_{ij} + [\mathbf{W}\mathbf{W}^{\mathrm{T}}\mathbf{V}\mathbf{V}^{\mathrm{T}}\mathbf{W}]_{ij} + [\mathbf{V}\mathbf{V}^{\mathrm{T}}\mathbf{W}\mathbf{W}^{\mathrm{T}}\mathbf{W}]_{ij}, \tag{35}$$

upon which the authors of [23] obtained the following multiplicative update rule using the technique in Section 2:

$$W_{ij}^{\text{new}} = W_{ij}\frac{2[\mathbf{V}\mathbf{V}^{\mathrm{T}}\mathbf{W}]_{ij}}{[\mathbf{W}\mathbf{W}^{\mathrm{T}}\mathbf{V}\mathbf{V}^{\mathrm{T}}\mathbf{W}]_{ij} + [\mathbf{V}\mathbf{V}^{\mathrm{T}}\mathbf{W}\mathbf{W}^{\mathrm{T}}\mathbf{W}]_{ij}}. \tag{36}$$

Similar to NMF, P-NMF is not the best method to minimize the reconstruction error. Instead, it focuses in training orthogonal basis vectors. The simulation results in [23] showed that P-NMF is capable of extracting highly localized, sparse, and non-overlapped part-based features. However, there is little explanation about this phenomenon in [23].

In this example we employ the reforming technique of the previous section to derive a new multiplicative update rule, named *Non-negative linear Hebbian network* (NLHN), for finding the non-negative projection. Given an $m$-dimensional non-negative input vector $\mathbf{v}$, we define the learning direction by the simplest linear Hebbian learning rule, i.e. the product of the input and the output of a linear network:

$$\mathbf{G} = \mathbf{v}(\mathbf{W}^{\mathrm{T}}\mathbf{v})^{\mathrm{T}} = \mathbf{v}\mathbf{v}^{\mathrm{T}}\mathbf{W}. \tag{37}$$

Inserting this to (28) with $\mathbf{v}\mathbf{v}^{\mathrm{T}} = \mathbf{A} = \mathbf{A}^{+}$, we obtain the following update rule:

$$W_{ij}^{\text{new}} = W_{ij}\frac{[\mathbf{v}\mathbf{v}^{\mathrm{T}}\mathbf{W}]_{ij}}{[\mathbf{W}\mathbf{W}^{\mathrm{T}}\mathbf{v}\mathbf{v}^{\mathrm{T}}\mathbf{W}]_{ij}}. \tag{38}$$

This result is tightly connected to the PCA approach because the corresponding additive update rule

$$\mathbf{W}^{\text{new}} = \mathbf{W} + \gamma(\mathbf{v}\mathbf{v}^{\mathrm{T}}\mathbf{W} - \mathbf{W}\mathbf{W}^{\mathrm{T}}\mathbf{v}\mathbf{v}^{\mathrm{T}}\mathbf{W}) \tag{39}$$

is a neural network implementation of PCA [18], which results in a set of eigenvectors for the largest eigenvalues of $E\{(\mathbf{v} - E\{\mathbf{v}\})(\mathbf{v} - E\{\mathbf{v}\})^{\mathrm{T}}\}$. However, these eigenvectors and the principal components of data found by PCA are not necessarily non-negative.

In addition to the on-line learning rule (38), we can also use its batch version

$$W_{ij}^{\text{new}} = W_{ij}\frac{[\mathbf{V}\mathbf{V}^{\mathrm{T}}\mathbf{W}]_{ij}}{[\mathbf{W}\mathbf{W}^{\mathrm{T}}\mathbf{V}\mathbf{V}^{\mathrm{T}}\mathbf{W}]_{ij}}, \tag{40}$$

where $\mathbf{V}$ is an $m \times n$ matrix, each column for one non-negative input sample.

We can see that NLHN bridges NMF and P-NMF. While the latter replaces $\mathbf{H}$ with $\mathbf{W}^{\mathrm{T}}\mathbf{V}$ in the NMF objective function (30), NLHN applies similar replacement in the update rule (32) of NMF. In addition, NLHN can be considered as a slight variant of P-NMF. To see this, let us decompose (35) into two parts

$$\frac{\partial \mathscr{J}_{\text{P-NMF}}(\mathbf{W})}{\partial W_{ij}} = [\mathbf{G}^{(1)}]_{ij} + [\mathbf{G}^{(2)}]_{ij}, \tag{41}$$

where $\mathbf{G}^{(1)} = -\mathbf{V}\mathbf{V}^{\mathrm{T}}\mathbf{W} + \mathbf{W}\mathbf{W}^{\mathrm{T}}\mathbf{V}\mathbf{V}^{\mathrm{T}}\mathbf{W}$ and $\mathbf{G}^{(2)} = -\mathbf{V}\mathbf{V}^{\mathrm{T}}\mathbf{W} + \mathbf{V}\mathbf{V}^{\mathrm{T}}\mathbf{W}\mathbf{W}^{\mathrm{T}}\mathbf{W}$. It has been shown that $\mathbf{G}^{(2)}$ has little effect in learning the principal directions [9]. Thus, by dropping $\mathbf{G}^{(2)}$, we obtain the same multiplicative update rule as (40) based on $\mathbf{G}^{(1)}$. Unlike other variants of NMF such as [5,14,22], NLHN does not require any additional regularization terms. This holds also for P-NMF.

The major novelty of NLHN does not lie in its performance as we have shown that it is essentially the same as P-NMF. We will also show by experiments in Section 6.2 that NLHN behaves very similarly to P-NMF. However, the interpretation of P-NMF as a variant of simple Hebbian learning with normalization helps us understand the underlying reason that P-NMF and NLHN are able to learn more localized and non-overlapped parts of the input samples.

As we know, iteratively applying the same Hebbian learning rule will result in that the winning neuron is repeatedly enhanced, and the normalization forces only one neuron to win all energy from the objective function [7]. In our case, this means that only one entry of each row of $\mathbf{W}$ will finally remain non-zero and the others will be squeezed to zero. That is, the normalized Hebbian learning is the underlying cause of the implicit orthogonalization. Furthermore, notice that two non-negative vectors are orthogonal if and only if their non-zero parts are not overlapped.

Then why can P-NMF or NLHN produce localized representations? To see this, one should first notice that the objectives of P-NMF and NLHN are essentially the same if $\mathbf{W}$ is orthonormal. Therefore, let us only consider here the objective of NLHN, $\mathscr{J}_{\text{NLHN}}(\mathbf{W}) = E\{\|\mathbf{W}^{\mathrm{T}}\mathbf{v}\|^2\}$, which can be interpreted as the mean correlation between projected vectors. In many applications, e.g. facial images, the correlation often takes place in neighboring pixels or symmetric parts. That is why P-NMF or NLHN can extract the correlated facial parts such as lips and eye brows, etc.

We can further deduce the generative model of P-NMF or NLHN. Assume each observation $\mathbf{v}$ is composed of $r$ non-overlapped parts, i.e. $\mathbf{v} = \sum_{p=1}^{r}\mathbf{v}_p$. In the context of orthogonality, P-NMF models each part $\mathbf{v}_p$ by the scaling of a basis vector $\mathbf{w}_p$ plus a noise vector $\boldsymbol{\varepsilon}_p$:

$$\mathbf{v}_p = \alpha_p\mathbf{w}_p + \boldsymbol{\varepsilon}_p. \tag{42}$$

If the basis vectors are normalized so that $\mathbf{w}_p^{\mathrm{T}}\mathbf{w}_q = 1$ for $q = p$ and 0 otherwise, then the reconstructed vector

of this part is

$$\sum_{q=1}^{r}\mathbf{w}_q\mathbf{w}_q^{\mathrm{T}}\mathbf{v}_p = \sum_{q=1}^{r}\mathbf{w}_q\mathbf{w}_q^{\mathrm{T}}(\alpha_p\mathbf{w}_p+\boldsymbol{\varepsilon}_p) = \sum_{q=1}^{r}\alpha_p\mathbf{w}_q\mathbf{w}_q^{\mathrm{T}}\mathbf{w}_p+\sum_{q=1}^{r}\mathbf{w}_q\mathbf{w}_q^{\mathrm{T}}\boldsymbol{\varepsilon}_p = \alpha_p\mathbf{w}_p+\sum_{q=1}^{r}\mathbf{w}_q\mathbf{w}_q^{\mathrm{T}}\boldsymbol{\varepsilon}_p. \quad (43)$$

The norm of the reconstruction error is therefore bounded by

$$\begin{aligned}\left\|\mathbf{v}_p-\sum_{q=1}^{r}\mathbf{w}_q\mathbf{w}_q^{\mathrm{T}}\mathbf{v}_p\right\| &= \left\|\left(\mathbf{I}-\sum_{q=1}^{r}\mathbf{w}_q\mathbf{w}_q^{\mathrm{T}}\right)\boldsymbol{\varepsilon}_p\right\| \\ &\leqslant \left\|\left(\mathbf{I}-\sum_{q=1}^{r}\mathbf{w}_q\mathbf{w}_q^{\mathrm{T}}\right)\right\|\cdot\|\boldsymbol{\varepsilon}_p\| \\ &= \mathrm{Tr}\left(\left(\mathbf{I}-\sum_{q=1}^{r}\mathbf{w}_q\mathbf{w}_q^{\mathrm{T}}\right)^{\mathrm{T}}\left(\mathbf{I}-\sum_{q=1}^{r}\mathbf{w}_q\mathbf{w}_q^{\mathrm{T}}\right)\right)\cdot\|\boldsymbol{\varepsilon}_p\| \\ &= \mathrm{Tr}\left(\mathbf{I}-\sum_{q=1}^{r}\mathbf{w}_q\mathbf{w}_q^{\mathrm{T}}\right)\cdot\|\boldsymbol{\varepsilon}_p\| \\ &= \left(\mathrm{Tr}(\mathbf{I})-\sum_{q=1}^{r}\mathrm{Tr}(\mathbf{w}_q^{\mathrm{T}}\mathbf{w}_q)\right)\cdot\|\boldsymbol{\varepsilon}_p\| \\ &= (m-r)\cdot\|\boldsymbol{\varepsilon}_p\| \end{aligned} \quad (44)$$

if 2-norm is used. Similar bounds can be derived for other types of norms. In words, $\mathbf{w}_p\mathbf{w}_p^{\mathrm{T}}\mathbf{v}_p$ reconstructs $\mathbf{v}_p$ well if the noise level $\boldsymbol{\varepsilon}_p$ is small enough. According to this model, P-NMF or NLHN can potentially be applied to signal processing problems where the global signals can be divided into several parts and for each part the observations mainly distribute along a straight line modeled by $\alpha_p\mathbf{w}_p$. This is closely related to Oja's PCA subspace rule [18], which finds the direction of the largest variation, except that the straight line found by P-NMF or NLHN has to pass through the origin.

It is important to notice that we analyze the forces of orthogonality and locality separately only for simplicity. Actually either P-NMF or NLHN implements them in the same multiplicative updates and these forces work concurrently to attain both goals during the learning.

### 5.2. Non-negative FDA

Given a set of non-negative multivariate samples $\mathbf{x}^{(i)}$, $i=1,\ldots,n$, from the vector space $\Re^m$, where the index of each sample is assigned to one of $Q$ classes, *Fisher* LDA finds the direction $\mathbf{w}$ for the following optimization problem:

$$\underset{\mathbf{w}\in\Re^m}{\text{maximize}} \quad \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\mathbf{w} \quad (45)$$

$$\text{subject to} \quad \mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w}=1. \quad (46)$$

Here $\mathbf{S}_{\mathrm{B}}$ is the *between classes scatter matrix* and $\mathbf{S}_{\mathrm{W}}$ is the *within classes scatter matrix*, defined as

$$\mathbf{S}_{\mathrm{B}} = \frac{1}{n}\sum_{c=1}^{Q}n_c(\boldsymbol{\mu}^{(c)}-\boldsymbol{\mu})(\boldsymbol{\mu}^{(c)}-\boldsymbol{\mu})^{\mathrm{T}}, \quad (47)$$

$$\mathbf{S}_{\mathrm{W}} = \frac{1}{n}\sum_{c=1}^{Q}\sum_{i\in\mathscr{I}_c}(\mathbf{x}^{(i)}-\boldsymbol{\mu}^{(c)})(\mathbf{x}^{(i)}-\boldsymbol{\mu}^{(c)})^{\mathrm{T}}, \quad (48)$$

where $n_c$ and $\mathscr{I}_c$ are the number and indices of the samples in class $c$, and

$$\boldsymbol{\mu}^{(c)} = \frac{1}{n_c}\sum_{i\in\mathscr{I}_c}\mathbf{x}^{(i)}, \quad (49)$$

$$\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}^{(i)} = \frac{1}{n}\sum_{c=1}^{Q}n_c\boldsymbol{\mu}^{(c)}. \quad (50)$$

If we set $\mathbf{v}=\mathbf{x}-\boldsymbol{\mu}$ and

$$E\{\mathbf{v}\mathbf{v}^{\mathrm{T}}\} \approx \mathbf{S}_{\mathrm{T}} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}^{(i)}-\boldsymbol{\mu})(\mathbf{x}^{(i)}-\boldsymbol{\mu})^{\mathrm{T}}, \quad (51)$$

and notice that the *total scatter matrix* $\mathbf{S}_{\mathrm{T}}=\mathbf{S}_{\mathrm{B}}+\mathbf{S}_{\mathrm{W}}$, the LDA problem can be reformulated as

$$\underset{\mathbf{w}\in\Re^m}{\text{maximize}} \quad \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{T}}\mathbf{w} \quad (52)$$

$$\text{subject to} \quad \mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w}=1, \quad (53)$$

and hence becomes a particular case of (9).

The common solution of LDA is to attach the constraint (46) to the objective function with a Lagrange multiplier, and then solve the *Karush–Kuhn–Tucker* (KKT) equation by SVD. This approach, however, fails to produce the non-negativity of $\mathbf{w}$. To overcome this, we apply the multiplicative update technique described in Section 4 and obtain the following novel alternative method here named *non-negative linear discriminant analysis* (NLDA).

We start from the steepest-gradient learning:

$$\mathbf{g}(\mathbf{w}) = \frac{\partial(\frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\mathbf{w})}{\partial\mathbf{w}} = \mathbf{S}_{\mathrm{B}}\mathbf{w}. \quad (54)$$

Because both $\mathbf{S}_{\mathrm{B}}$ and $\mathbf{S}_{\mathrm{W}}$ are symmetric, we have $\mathbf{A}=\mathbf{S}_{\mathrm{B}}$ and $\mathbf{B}=\mathbf{S}_{\mathrm{W}}$ and (21) becomes

$$w_i^{\mathrm{new}} = w_i\frac{[\mathbf{S}_{\mathrm{B}}^{+}\mathbf{w}]_i}{[\mathbf{S}_{\mathrm{B}}^{-}\mathbf{w}]_i+[\mathbf{w}\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{S}_{\mathrm{B}}\mathbf{w}]_i}, \quad (55)$$

with the elements of $\mathbf{w}$ initialized with random values from $(0,1)$ and $\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w}=1$ fulfilled. Here $\mathbf{S}_{\mathrm{B}}^{+}$ and $\mathbf{S}_{\mathrm{B}}^{-}$ are determined by (16). If the symmetric matrix $\mathbf{S}_{\mathrm{W}}\mathbf{S}_{\mathrm{B}}+(\mathbf{S}_{\mathrm{W}}\mathbf{S}_{\mathrm{B}})^{\mathrm{T}}=\mathbf{S}_{\mathrm{W}}\mathbf{S}_{\mathrm{B}}+\mathbf{S}_{\mathrm{B}}\mathbf{S}_{\mathrm{W}}$ is positive definite, then

$$\begin{aligned}\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{S}_{\mathrm{B}}\mathbf{w} &= \tfrac{1}{2}(\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{S}_{\mathrm{B}}\mathbf{w}+(\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{S}_{\mathrm{B}}\mathbf{w})^{\mathrm{T}}) \\ &= \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}(\mathbf{S}_{\mathrm{W}}\mathbf{S}_{\mathrm{B}}+\mathbf{S}_{\mathrm{B}}\mathbf{S}_{\mathrm{W}})\mathbf{w}>0. \end{aligned} \quad (56)$$

That is, all terms on the right side of (55) are positive and so is the new $\mathbf{w}$ after each iteration. Moreover, notice that NLDA does not require matrix inversion of $\mathbf{S}_{\mathrm{W}}$, an operation which is computationally expensive and possibly leads to singular problems.

For a discriminant analysis of two classes $C$ and $\bar{C}$, it is possible to improve the non-negative learning by preprocessing the data. For a dimension $j$, if median$\{x_j|x_j$

$\in C\} > \text{median}\{x_j | x_j \in \bar{C}\}$, we transform the original data by

$$x_j \leftarrow \Theta_j - x_j, \tag{57}$$

where $\Theta_j$ is the known upper bound of the $j$th dimension. After such flipping, for each dimension $j$ there exists a threshold $\theta_j$ which is larger than more than half of the samples of $C$ and smaller than more than half of the samples of $\bar{C}$. That is, the class $C$ mainly distributes the inner part closer to the origin while $\bar{C}$ distributes the outer part farther from the origin. Projecting the samples to the direction obtained by (55) will thus yield better discrimination.

The above non-negative discriminant analysis can be extended to non-linear case by using the kernel technique. Let $\mathbf{\Phi}$ be a mapping from $\Re^m$ to $\mathscr{F}$, which is implicitly defined by a kernel function $k(\mathbf{x},\mathbf{y}) = k(\mathbf{y},\mathbf{x}) = \mathbf{\Phi}(\mathbf{x})^{\mathrm{T}}\mathbf{\Phi}(\mathbf{y})$. *kernel Fisher discriminant analysis* (KFDA) [2,15] finds a direction $\mathbf{w}$ in the mapped space $\mathscr{F}$, which is the solution of the optimization problem

$$\underset{\mathbf{w}\in\Re^m}{\text{maximize}} \quad \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}^{\mathbf{\Phi}}\mathbf{w} \tag{58}$$

$$\text{subject to} \quad \mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{T}}^{\mathbf{\Phi}}\mathbf{w} = 1, \tag{59}$$

where $\mathbf{S}_{\mathrm{B}}^{\mathbf{\Phi}}$ and $\mathbf{S}_{\mathrm{W}}^{\mathbf{\Phi}}$ are the corresponding between-class and within-class scatter matrices. We use here $\mathbf{S}_{\mathrm{T}}^{\mathbf{\Phi}} = \mathbf{S}_{\mathrm{B}}^{\mathbf{\Phi}} + \mathbf{S}_{\mathrm{W}}^{\mathbf{\Phi}}$ instead of $\mathbf{S}_{\mathrm{W}}^{\mathbf{\Phi}}$ for simplification, but it is easy to see that the problems are equivalent. From the theory of reproducing kernels [20] we know that any solution $\mathbf{w} \in \mathscr{F}$ must lie in the span of all training samples in $\mathscr{F}$. That is, there exists an expansion for $\mathbf{w}$ of the form

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i \mathbf{\Phi}(\mathbf{x}^{(i)}). \tag{60}$$

It has been shown [2] that by substituting (60) into (58) and (59), the unconstrained KFDA can be expressed as

$$\underset{\boldsymbol{\alpha}\in\Re^n}{\text{maximize}} \quad \tfrac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{K}\mathbf{U}\mathbf{K}\boldsymbol{\alpha} \tag{61}$$

$$\text{subject to} \quad \boldsymbol{\alpha}^{\mathrm{T}}\mathbf{K}\mathbf{K}\boldsymbol{\alpha} = 1, \tag{62}$$

where $K_{ij} = k(\mathbf{x}^{(i)},\mathbf{x}^{(j)})$, $\mathbf{U} = \text{diag}(\mathbf{U}^{(1)},\mathbf{U}^{(2)},\ldots,\mathbf{U}^{(Q)})$, and $\mathbf{U}^{(j)}$ is an $n_j \times n_j$ matrix whose elements are $1/n_j$.

The matrices $\mathbf{K}$ and $\mathbf{U}$ are symmetric and all their elements are non-negative. We can obtain the following multiplicative update rule for novel *non-negative kernel Fisher discriminant analysis* (NKFDA) by setting $\mathbf{A} = \mathbf{A}^+ = \mathbf{K}\mathbf{U}\mathbf{K}$ and $\mathbf{B} = \mathbf{K}\mathbf{K}$ in (21):

$$\alpha_i^{\text{new}} = \alpha_i \frac{[\mathbf{K}\mathbf{U}\mathbf{K}\boldsymbol{\alpha}]_i}{[\boldsymbol{\alpha}\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{K}\mathbf{K}\mathbf{K}\mathbf{U}\mathbf{K}\boldsymbol{\alpha}]_i}. \tag{63}$$

This formulation has the extra advantage that the resulting elements of $\boldsymbol{\alpha}$ indicate the contribution of their respective samples in forming the discriminative projection. This is conducive to selecting among the samples and revealing the underlying factor in the data even if we only use the simple linear kernel, i.e. $\mathbf{\Phi}(\mathbf{x}) = \mathbf{x}$.

## 6. Experiments

We demonstrate here the empirical results of the non-negative algorithms presented in Sections 5.1 and 5.2 when applied for processing of facial images. Before proceeding to details, it should be emphasized that the goal of the non-negative version of a given algorithm usually differs from the original one. The resulting objective value of a non-negative algorithm generally is not as good as that of its unconstrained counterpart. However, readers should be aware of that data analysis is not restricted to reconstruction or classification and that non-negative learning can bring us novel insights in the data.

### 6.1. Data

We have used the FERET database of facial images [19]. After face segmentation, 2409 frontal images (poses "fa" and "fb") of 867 subjects were stored in the database for the experiments. We obtained the coordinates of the eyes from the ground truth data of FERET collection, with which we calibrated the head rotation so that all faces are upright. Afterwards, all face boxes were normalized to the size of $32 \times 32$, with fixed locations for the left eye (26,9) and the right eye (7,9). Each image was reshaped to a 1024-dimensional vector by column-wise concatenation.

Another database we used is the UND database (collection B) [6], which contains 33,247 frontal facial images. We applied the same preprocessing procedure to the UND images as to the FERET database.

### 6.2. Non-negative linear Hebbian networks

First we compared four unsupervised methods: PCA, NMF [12], P-NMF [23] and NLHN (40) for encoding the faces. The resulting basis images are shown in Fig. 1. It can be seen that the PCA bases are holistic and it is hard to identify the parts that compose a face. NMF yields partially localized features of a face, but some of them are still heavily overlapped. P-NMF and NLHN are able to extract the highly sparse, local and non-overlapped parts of the face, for example the nose and the eyebrows. The major difference between P-NMF and NLHN is only the order of the basis vectors.

Orthogonality is of main interests for part-based learning methods because that property leads to non-overlapped parts and localized representations as discussed in Section 5.1. Suppose the normalized inner product between two basis vectors $\mathbf{w}_i$ and $\mathbf{w}_j$ is

$$R_{ij} = \frac{\mathbf{w}_i^{\mathrm{T}}\mathbf{w}_j}{\|\mathbf{w}_i\|\|\mathbf{w}_j\|}. \tag{64}$$

Then the orthogonality of an $m \times r$ basis $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_r]$ can be quantified by the following $\rho$-measurement:

$$\rho = 1 - \frac{\sum_{i \neq j} R_{ij}}{r(r-1)} \tag{65}$$

so that $\rho \in [0, 1]$. Larger $\rho$'s indicate higher orthogonality and $\rho$ reaches 1 when the columns of $\mathbf{W}$ are completely orthogonal.

Fig. 2 shows the evolution of $\rho$'s by using NMF, NLHN and P-NMF with dimensions $m = 1024$ and $r = 25$. NMF converges to a local minimum with $\rho = 0.63$, while NLHN and P-NMF learn $\mathbf{W}$ with $\rho = 0.97$ and 0.98, respectively, after 5000 iterations. We also trained NLHN and P-NMF with different random seeds for the initial values of $\mathbf{W}$ and the results are shown in Fig. 3. It can be seen that both methods converge with very similar curves. That is, NLHN behaves very similarly to P-NMF in attaining high orthogonality, and being not sensitive to the initial values.

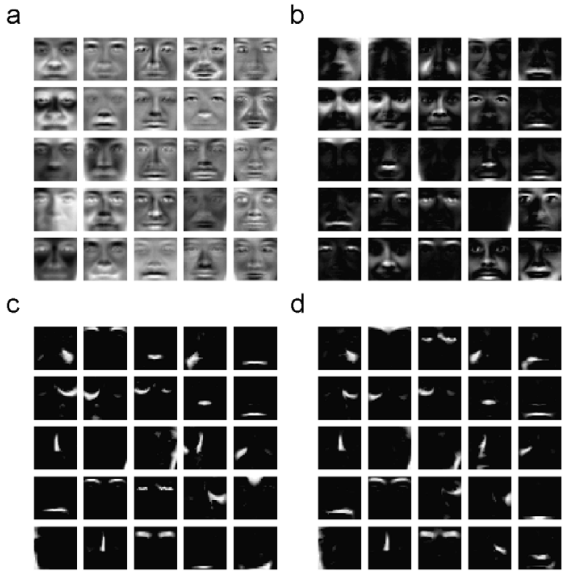


Fig. 1. The top 25 learned bases of (a) PCA, (b) NMF, (c)P-NMF and (d) NLHN. All base images have been plotted without any extra scaling.

### 6.3. Non-negative linear discriminant analysis

Next, we demonstrate the application of the linear supervised non-negative learning in discriminating whether the person in a given image has mustache and whether he or she is wearing glasses. The data were preprocessed by (57) before applying the NLDA algorithm (55). The positive definity requirement (56) was checked to be fulfilled for these two supervised learning problems. In addition to LDA, we chose the *linear support vector machine* (L-SVM) [3] for comparison. Suppose $n$ samples $\mathbf{x}^{(i)}$ labeled by $y^{(i)} \in \{1, -1\}$, $i = 1, 2, \ldots, n$. L-SVM seeks the solution of the optimization problem

$$\underset{\mathbf{w} \in \Re^m}{\text{maximize}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi^{(i)} \tag{66}$$

$$\text{subject to} \quad y^{(i)}(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(i)} + b) \geqslant 1 - \xi^{(i)} \tag{67}$$

and

$$\xi^{(i)} \geqslant 0, \quad i = 1, 2, \ldots, n, \tag{68}$$

where $C$ is a user-provided tradeoff parameter. By solving its dual problem

$$\underset{\boldsymbol{\alpha}}{\text{maximize}} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y^{(i)} y^{(j)} \mathbf{x}^{(i)} \mathbf{x}^{(j)} \tag{69}$$

$$\text{subject to} \quad \sum_{i=1}^{n} \alpha_i y^{(i)} = 0 \quad \text{and} \quad 0 \leqslant \alpha_i \leqslant C, \quad i = 1, 2, \ldots, n, \tag{70}$$

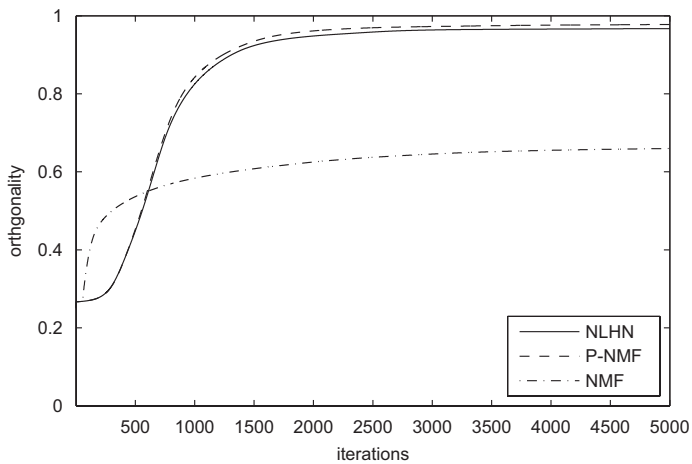


Fig. 2. $\rho$ values of NLHN, P-NMF and NMF with 25 basis vectors and 1024-dimensional data.

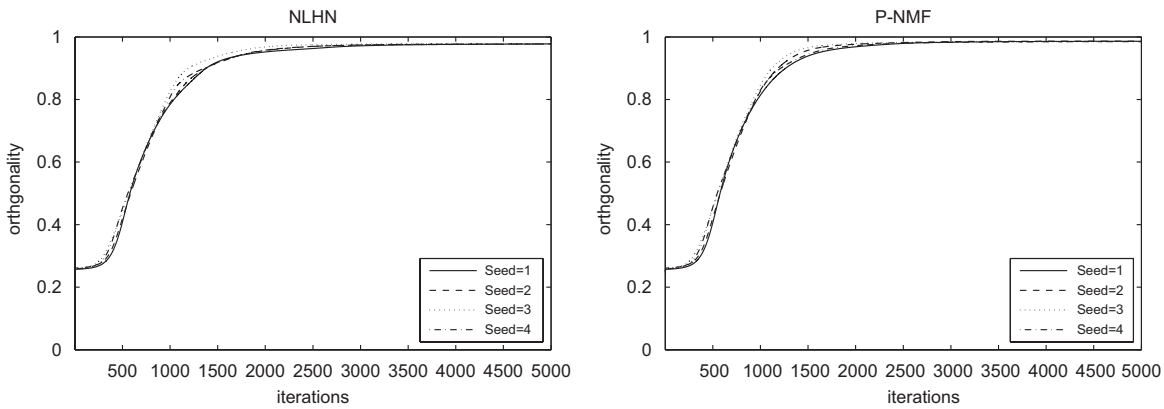


Fig. 3. $\rho$ values of NLHN and P-NMF with different random seeds.

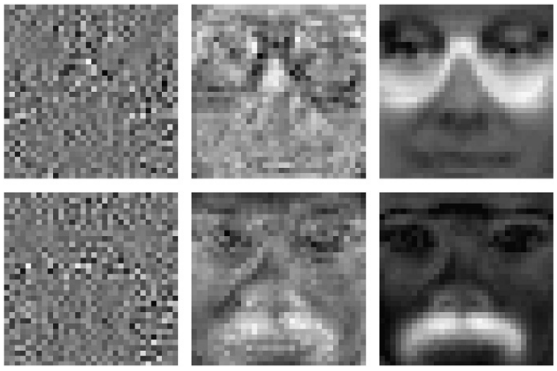

Fig. 4. Images of the projection vector for discriminating glasses (the first row) and mustache (the second row). The methods used from left to right are LDA, L-SVM, NLDA.

one obtains the optimal $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y^{(i)} \mathbf{x}^{(i)}$ with respect to the given $C$.

Fig. 4 displays the respective resulting LDA, L-SVM and NLDA projection vectors as images. It is hard to tell from the LDA projection vector in which dimensions of the input samples are most relevant for discriminating mustache or glasses. Such poor filters are probably caused by overfitting because LDA may require large amount of data to approximate the within-class covariance by $\mathbf{S}_W$ but this becomes especially difficult for high-dimensional problems. The filter images of L-SVM are slightly better than those of LDA. A distinguishing part can roughly be perceived between the eyes. Nevertheless there are still many irrelevant points found by L-SVM. In contrast, the NLDA training clearly extract the relevant parts, as shown by the brightest pixels in the rightmost images in Fig. 4. Such results conform well to the human intuition in this task.

Next we compared the above methods in classification where the training data were the FERET database and test data were the UND database. The compared methods try to classify whether the person in the image wears glasses or not because both data sets contain this kind of ground truth data. (Unfortunately the UND database does not include such data on mustache.) The results are shown in Table 1, from which we can see that LDA performs best in classifying the training samples, but very poorly for the new data outside of the training set.

Table 1
*Equal error rates* (EERs) of training data (FERET) and test data (UND) in classification of glasses versus no glasses

| | LDA | L-SVM | NLDA |
|---|---|---|---|
| FERET (%) | 0.98 | 11.69 | 17.54 |
| UND (%) | 32.35 | 25.23 | 23.88 |

L-SVM requires choice of the tradeoff parameter $C$. To our knowledge, there are no efficient and automatic methods for obtaining its optimal value. In this experiment we trained different L-SVMs with the tradeoff parameter in $\{0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30, 100\}$ and it turned out that the one with 0.03 performed best in fivefold cross-validations. Using this value, we trained an L-SVM with all training data and ran the classification on the test data. The result shows that L-SVM generalizes better than LDA, but the test error rate is still much higher than the training one, which is possibly because of some overfitting dimensions in the L-SVM filter. Higher classification accuracy could be obtained by applying non-linear kernels. This, however, requires more effort to choose among kernels and associated kernel parameters, which is beyond the scope of this paper.

The last column of Table 1 shows that NLDA performed even better than L-SVM in classifying the test samples. This is possibly because of the variation between the two databases. In such a case, the NLDA filter, which resembles more our prior knowledge, tends to be more reliable.

### 6.4. Non-negative kernel Fisher discriminant analysis

Furthermore, we tested the function of NKFDA (63) that ranks the training samples by their contribution to the projection. We used only the linear kernel here because of its simplicity. After training of NKFDA on the FERET data labeled whether the subject has mustache, we sorted the training samples in the order of their respective values of $\alpha_i$.

The top 25 faces and the bottom 25 faces are shown in Figs. 5(a) and (b), respectively. It can be clearly seen that the most important factor is the lighting condition. This could be expected because we are using well-aligned facial images of front-pose and neutral expression. Therefore the most significant noise here can be assumed to be variation in lighting. In addition, this order agrees with the common perception of humans, where the lighting that provides enough contrast helps in discriminating semantic classes.

For comparison, we display the ordered results of two compared methods. Figs. 5(c) and (d) show the top 25 and bottom 25 faces of KFDA, respectively. Because the coefficients produced by KFDA are not necessarily non-negative, we sorted the faces by their absolute values. The bottom images look similar to those obtained by NKFDA. That is, darker images that provide poor contrast contribute least to the discrimination. However, it is hard to find a common condition or easy interpretation for the KFDA ranking of the top faces.

The resulting top and bottom facial images of L-SVM are shown in Figs. 5(e) and (f). As we know, the samples with non-zero coefficients are support vectors, i.e. those around the classification boundary, which explains the top images ranked by L-SVM. On the other hand, the samples far away from the boundary will be associated with zero coefficients. In this case, they are mostly typical non-mustache faces shown in Fig. 5(f).

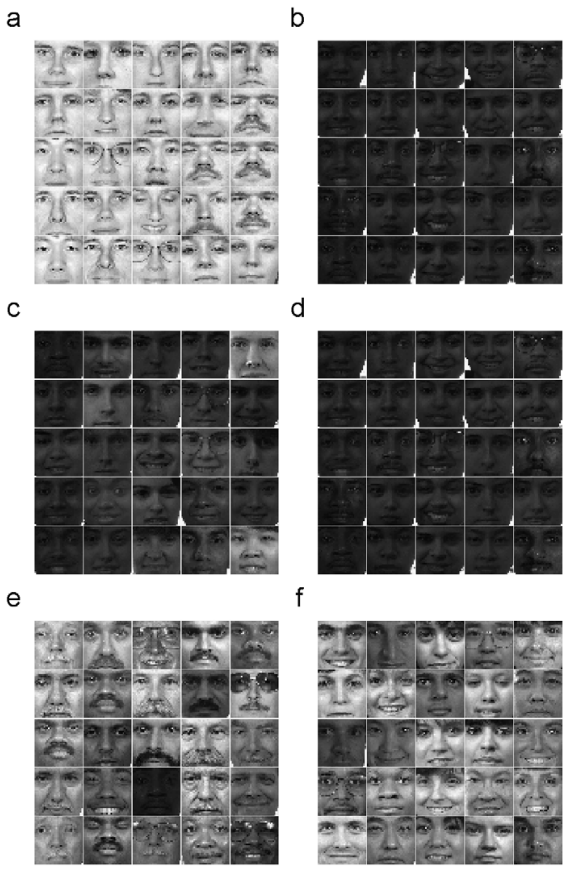


Fig. 5. The images with: (a) largest and (b) smallest values in $\boldsymbol{\alpha}$ trained by NKFDA; (c) largest and (d) largest absolute values in $\boldsymbol{\alpha}$ trained by KFDA; (e) largest and (f) smallest values in $\boldsymbol{\alpha}$ trained by L-SVM.

## 7. Conclusions

We presented a technique how to construct multiplicative updates for learning non-negative projections based on Oja's rule, including two examples of its application to reforming the conventional PCA and FDA to their non-negative variants. In the experiments on facial images, the non-negative projections learned by using the novel iterative update rules have demonstrated interesting properties for data analysis tasks in and beyond reconstruction and classification.

It is still a challenging and open problem to mathematically prove the convergence of orthogonality. The derivation of our method provides a possible interpretation of the multiplicative updates. The numerator part favors the learning in the original direction while the denominator part is mainly responsible for the normalization constraint. These two forces work together to steer the learning towards a local optimum. Iteratively applying such a multiplicative update rule yields one of the two sorts of stationarity in each dimension, one approaching zero and the other reaching the natural upper bound. Note that none of the elements can approach infinity because of the normalization. In the terms of neural networks, this can be interpreted as a competition between the elements, both within a neuron and across the neurons. In the matrix case, more than one neuron compete for the energy from the objective function and only one of them wins all over the others, which leads to high orthogonality between the learned basis vectors.

## References

[1] S. Amari, Natural gradient works efficiently in learning, Neural Comput. 10 (2) (1998) 251–276.
[2] G. Baudat, F. Anouar, Generalized discriminant analysis using a kernel approach, Neural Comput. 12 (10) (2000) 2385–2404.
[3] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines, Cambridge University Press, Cambridge, 2000.
[4] A. Edelman, The geometry of algorithms with orthogonality constraints, SIAM J. Matrix Anal. Appl. 20 (2) (1998) 303–353.

[5] T. Feng, S.Z. Li, H.Y. Shum, H.J. Zhang, Local non-negative matrix factorization as a visual representation, in: Proceedings, The Second International Conference on Development and Learning, 2002, pp. 178–183.
[6] P.J. Flynn, K.W. Bowyer, P.J. Phillips, Assessment of time dependency in face recognition: an initial study, Audio- and Video-Based Biometric Person Authentication, 2003, pp. 44–51.
[7] S. Haykin, Neural Networks—A Comprehensive Foundation, second ed., Prentice-Hall, Englewood Cliffs, NJ, 1998.
[8] A. Hyvärinen, P. Hoyer, Emergence of phase- and shift-invariant features by decomposition of natural images into independent feature subspaces, Neural Comput. 12 (7) (2000) 1705–1720.
[9] J. Karhunen, J. Joutsensalo, Generalizations of principal component analysis, optimization problems, and neural networks, Neural Networks 8 (4) (1995) 549–562.
[10] J. Kivinen, M. Warmuth, Exponentiated gradient versus gradient descent for linear predictors, Inf. Comput. 132 (1) (1997) 1–63.
[11] T. Kohonen, Emergence of invariant-feature detectors in the adaptive-subspace self-organizing map, Biol. Cybern. 75 (1996) 281–291.
[12] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (1999) 788–791.
[13] D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, in: NIPS, 2000, pp. 556–562.
[14] W. Liu, N. Zheng, X. Lu, Non-negative matrix factorization for visual coding, in: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2003), vol. 3, 2003, pp. 293–296.
[15] S. Mika, G. Rtsch, J. Weston, B. Schölkopf, K.-R. Mller, Fisher discriminant analysis with kernels, Neural Networks Signal Process. IX (1999) 41–48.
[16] Y. Nishimori, S. Akaho, Learning algorithms utilizing quasi-geodesic flows on the Stiefel manifold, Neurocomputing 67 (2005) 106–135.
[17] E. Oja, A simplified neuron model as a principal component analyzer, J. Math. Biol. 15 (1982) 267–273.
[18] E. Oja, Principal components, minor components, and linear neural networks, Neural Networks 5 (1992) 927–935.
[19] P.J. Phillips, H. Moon, S.A. Rizvi, P.J. Rauss, The FERET evaluation methodology for face recognition algorithms, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 1090–1104.
[20] B. Schölkopf, A. Smola, Learning with Kernels, MIT Press, Cambridge, MA, 2002.
[21] F. Sha, L.K. Saul, D.D. Lee, Multiplicative updates for large margin classifiers, in: COLT, 2003, pp. 188–202.
[22] B.W. Xu, J.J. Lu, G.S. Huang, A constrained non-negative matrix factorization in information retrieval, in: Proceedings of the 2003 IEEE International Conference on Information Reuse and Integration (IRI2003), 2003, pp. 273–277.
[23] Z. Yuan, E. Oja, Projective nonnegative matrix factorization for image compression and feature extraction, in: Proceedings of 14th Scandinavian Conference on Image Analysis (SCIA 2005), Joensuu, Finland, June 2005, pp. 333–342.
[24] S. Zafeiriou, A. Tefas, I. Buciu, I. Pitas, Class-specific discriminant non-negative matrix factorization for frontal face verification, in: Proceedings of Third International Conference on Advances in Pattern Recognition (ICAPR 2005), vol. 2, 2005, pp. 206–215.

**Zhirong Yang** received his Bachelor and Master degrees in Computer Science from Sun Yat-Sen University, Guangzhou, China, in 1999 and 2002, respectively. Presently he is a doctoral candidate at the Computer and Information Science Laboratory in Helsinki University of Technology. His research interests include machine learning, pattern recognition, computer vision, and multimedia retrieval.

**Jorma Laaksonen** received his Dr. of Science in Technology degree in 1997 from Helsinki University of Technology, Finland, where he is presently Academy Research Fellow of Academy of Finland at the Laboratory of Computer and Information Science. He is an author of several journal and conference papers on pattern recognition, statistical classification, and neural networks. His research interests are in content-based information retrieval and recognition of handwriting. Dr. Laaksonen is an IEEE senior member, a founding member of the SOM and LVQ Programming Teams and the PicSOM Development Group, and a member of the International Association of Pattern Recognition (IAPR) Technical Committee 3: Neural Networks and Machine Learning.