Weighted Nonnegative Matrix Factorization and Face Feature Extraction

Vincent D. Blondel\textsuperscript{a} Ngoc-Diep Ho\textsuperscript{a} Paul van Dooren\textsuperscript{a}

\textsuperscript{a}CESAME, Universit\'e catholique de Louvain, Av. Georges Lema\^{i}tre 4, B-1348 Louvain-la-Neuve, Belgium.

Abstract

In this paper we consider weighted nonnegative matrix factorizations and we show that the popular algorithms of Lee and Seung can incorporate such a weighting. We then prove that for appropriately chosen weighting matrices, the weighted Euclidean distance function and the weighted generalized Kullback-Leibler divergence function are essentially identical. We finally show that the weighting can be chosen to emphasize parts of the data matrix to be approximated and we can apply this to the low rank fitting of a face image database.

Key words: Non-negative matrix factorization, weighting, Euclidean distance, generalized Kullback-Leibler divergence

1 Introduction

Nonnegative Matrix Factorizations (NMF’s) are popular for the problem of approximating nonnegative data. The classical example is that of approximating a given image by a linear combination of other “parts” (i.e. simpler images) with the additional constraint that all images must be represented by a matrix with nonnegative elements: each matrix element gives the grey level of an image pixel, and is constrained to be nonnegative.

If the simpler images are nonnegative matrices of rank one then they can be written as a product $u_i v_i^T$ where both $u_i$ and $v_i$ are nonnegative vectors of
appropriate length. The approximation problem of a \( m \times n \) matrix \( A \) by a linear combination of \( k < m, n \) such products then reduces to

\[
A \approx \sum_{i=1}^{k} \sigma_i u_i v_i^T
\]

where the nonnegative elements \( \sigma_i \) are the weighting factors of the linear combination. When there is no constraint on the vectors \( u_i \) and \( v_i \) it is well known that the best rank \( k \) approximation in the Euclidean norm can be obtained via the Singular Value Decomposition, that is, \( \min \| A - \sum_{i=1}^{k} \sigma_i u_i v_i^T \| \) is achieved for \( u_i, v_i \) such that \( u_i^T u_j = 0 \) and \( v_i^T v_j = 0 \), \( \forall i \neq j \) and \( u_i^T u_i = v_i^T v_i = 1 \), \( \forall i \). Moreover there are good algorithms available to compute the optimal approximation in a computing time that is cubic in the dimensions \( m \) and \( n \) of the matrix \( A \) [5]. In many applications, nonnegativity of elements of \( u_i \) and \( v_i \) is a crucial property that one wants to preserve. Imposing the nonnegativity constraint makes the low-rank approximation problem non convex and much more difficult to solve. Lee and Seung have proposed [8] an efficient algorithm for obtaining such a approximation. In this paper, we describe how a weight matrix can be incorporated in the popular algorithms of Lee and Seung. This modification was first presented in two independent reports [1] and [3]. We then show in Section 4 that for appropriately chosen weighting matrices, the weighted Euclidean distance function and the weighted generalized Kullback-Leibler divergence function become essentially identical.

The first application of the Lee and Seung algorithms was to extract face features from a face database [8]. Thanks to the nonnegativity, each extracted feature was again an image and each face was reconstructed by a nonnegative combination of those features. But it turned out that, in stead of parts of faces, the original algorithms provide us with only some fuzzy faces as features. Sparsity constraints were then added to the algorithms [7] to obtain sparse and local features. Section 5 describes how the weighted approximation can be used to emphasize certain parts of faces with or without sparsity constraints.

2 Nonnegative Matrix Factorization

The Nonnegative Matrix Factorization problem can be stated as follows: Given a nonnegative \((m \times n)\) matrix \( A \), find two nonnegative matrices \( U (m \times k) \) and \( V (k \times n) \) that minimize \( F(A, UV) \), where \( F(A, B) \) is a cost function defining the “distance” between the matrices \( A \) and \( B \).

The choice of the cost function \( F \) of course affects the solution of the minimization problem. One popular choice is the Euclidean Distance (or the Frobenius
Another popular choice in image approximation problems is the Generalized Kullback-Leibler Divergence

\[ D(A \parallel UV) := \sum_{ij} \left[ A \circ \log \left( \frac{A}{UV} \right) - A + UV \right]_{ij}, \tag{2} \]

where \( \log(X) \) is the element-wise logarithm of \( X \), \( X \circ Y \) is the Hadamard product (or element by element product) of the matrices \( X \) and \( Y \), and \( \frac{X}{Y} \) is the Hadamard division (or element by element division) of the matrices \( X \) and \( Y \).

In [8,9], Lee and Seung propose two algorithms for finding local minimizers of these two cost functions. The algorithms are based on multiplicative updating rules which are simple but quite elegant. We will derive below two similar algorithms for the problem of Weighted Nonnegative Matrix Factorization (WNMF) which minimize the following weighted cost functions: the Weighted Euclidean Distance

\[ \frac{1}{2} \| A - UV \|^2_W := \frac{1}{2} \sum_{ij} [W \circ (A - UV) \circ (A - UV)]_{ij}, \tag{3} \]

and the Weighted Generalized Kullback-Leibler Divergence

\[ D_W(A \parallel UV) := \sum_{ij} \left[ W \circ \left( A \circ \log \left( \frac{A}{UV} \right) - A + UV \right) \right]_{ij}, \tag{4} \]

where \( W = \{W_{ij}\} > 0 \) is a nonnegative weight matrix. Clearly, the two earlier versions are just particular cases of the weighted ones where all the weights are equal to 1.

The problem of Weighted Nonnegative Matrix Factorization was first stated in [12] for the Weighted Euclidean Distance (3). Several algorithms including Newton-related methods were used to solve the problem, but they have a high complexity. Simpler algorithms were introduced by Lee and Seung [8,9] based on a set of multiplicative updating rules but these algorithms were presented for the unweighted Euclidean Distance and generalized KL Divergence.

Recently [6], a particular type of weighting was proposed for the divergence cost function, in order to vary the importance of each column of the matrix \( A \) in the approximation \( UV \approx AD \), where \( D \) is a nonnegative diagonal scaling matrix. One can easily see that this nonnegative weight matrix is equivalent to a rank-one weighting matrix \( W \) in our weighted generalized KL divergence.
An approach that allows to use weighting matrices in a more general context is given in [13], where an Expectation-Maximization algorithm is used in an iterative scheme that produces an unweighted low-rank approximation of a weighted combination of a previously computed approximation:

\[(U_{k+1}, V_{k+1}) = \text{LowRank}(W \circ A + (1 - W) \circ (U_k V_k)).\] (5)

Here there are no constraints of non-negativity, but the same idea can also be used to incorporate weights in an algorithm for nonnegative matrix factorizations. This implies that one has to solve an unweighted low-rank nonnegative approximation at each step of the iteration, and this can become quite inefficient in terms of complexity.

3 The Lee and Seung approach

We first briefly recall in this section the basic ideas of the Lee-Seung approach. Although the cost functions \(\frac{1}{2}\|A - UV\|^2\) and \(D(A\|UV)\) are not convex in the two matrix variables \(U\) and \(V\) (one can show that there are many local minimizers), it has been shown that for a fixed \(U\) the cost function is convex in \(V\), and vice-versa. A simple strategy to find a local minimizer is therefore to alternate between minimizations in \(U\) and \(V\) while keeping the other matrix fixed.

The minimization of \(F(A, U V)\) under the constraints \(U, V \geq 0\), requires the construction of the gradients \(\nabla_U\) and \(\nabla_V\) of the cost function \(F(A, U V)\). For the Euclidean Distance, these are:

\[\nabla_U \frac{1}{2} \|A - UV\|^2 = -(A - UV)V^T,\] (6)
\[\nabla_V \frac{1}{2} \|A - UV\|^2 = -U^T(A - UV).\] (7)

For the generalized KL divergence, the gradients are also easy to construct:

\[\nabla_U D(A\|UV) = -\left(\frac{[A]}{[UV]} - 1_{m\times n}\right)V^T,\] (8)
\[\nabla_V D(A\|UV) = -U^T\left(\frac{[A]}{[UV]} - 1_{m\times n}\right)\] (9)

where \(1_{m\times n}\) is a \(m \times n\) matrix with all elements equal to 1.

For the two cost functions, the Kuhn-Tucker conditions are then:

\[U \geq 0, \quad V \geq 0,\] (10)
\[ \nabla_U F(A, UV) \geq 0, \quad \nabla_V F(A, UV) \geq 0, \quad (11) \]
[ \nabla_U F(A, UV) = 0, \quad V \circ \nabla_V F(A, UV) = 0, \quad (12) \]

where \( F(A, UV) \) is either \( \frac{1}{2} \| A - UV \|_2^2 \) or \( D(A\|UV) \).

Lee and Seung [8] propose simple updating rules to minimize the cost function. Their convergence results are described in the following two theorems [8,9]:

**Theorem 1** The Euclidean distance \( \frac{1}{2} \| A - UV \|_2^2 \) is non-increasing under the updating rules:

\[
V \leftarrow V \circ \frac{[U^T A]}{U^T U V}, \quad U \leftarrow U \circ \frac{[AV^T]}{UV V^T}. \quad (13)
\]

The Euclidean distance \( \frac{1}{2} \| A - UV \|_2^2 \) is invariant under these updates iff \( U \) and \( V \) are at a stationary point of the distance.

**Theorem 2** The divergence \( D(A\|UV) \) is non-increasing under the updating rules:

\[
V \leftarrow \frac{[V]}{[U^T 1_{m\times n}]} \circ \left( \frac{[A]}{[U]} \frac{[U V]}{[U V]} \right), \quad U \leftarrow \frac{[U]}{[1_{m\times n} V^T]} \circ \left( \frac{[A]}{[U V]} V^T \right), \quad (14)
\]

where \( 1_{m\times n} \) is a \( m \times n \) matrix with all elements equal to 1. The divergence \( D(A\|UV) \) is invariant under these updates iff \( U \) and \( V \) are at a stationary point of the divergence.

The above updating rules are the same as in [8,9] but are rewritten here in matrix form using the Hadamard product and Hadamard division, in order to allow an easy comparison with the updating rules for the weighted cases. The proofs of these theorems can be found in [9], and will be extended for the weighted cases in the next section. The claims for stationary point in [9] may not always hold since the authors only showed that there exist limit points satisfying the conditions (10) and (12). These updating rules do not reveal whether the conditions (11) hold. In fact, these updating rules only guarantee non-increasing updates but do not guarantee a convergence to a local minimum. But in practice, they do produce satisfactory results in many applications. Some further remarks can be made about these algorithms:

**Remark 1:** The nonnegativity constraint on the matrices \( U \) and \( V \) is automatically satisfied by these updating rules if the starting matrices \( U_0 \) and \( V_0 \) are nonnegative.

**Remark 2:** In order to prevent divisions by zero due to rounding errors during the execution of the algorithm, we replace in practice, the above updating rules by the following ones:

\[
V \leftarrow V \circ \frac{[U^T A]}{[U^T U V + \epsilon 1_{k\times n}]}, \quad U \leftarrow U \circ \frac{[AV^T]}{[UV V^T + \epsilon 1_{m\times k}]} \quad (15)
\]
for the Euclidean Distance and
\[ V \leftarrow \frac{[V]}{[U^T 1_{m \times n}]} \odot \left( U^T \frac{[A]}{[UV + \epsilon 1_{m \times n}]} \right), \]
\[ U \leftarrow \frac{[U]}{[1_{m \times n} V^T]} \odot \left( \frac{[A]}{[UV + \epsilon 1_{m \times n}]} V^T \right) \]  
(16)

for the Generalized Kullback-Leibler divergence, where \( \epsilon \) is a small positive constant. A particular modification of these updating rules was analyzed in [11].

4 Weighted Nonnegative Matrix Factorization

In this section we extend the results of Lee and Seung to the weighted case. We treat the different cases separately.

4.1 The weighted Euclidean distance

In order to generalize Theorem 1 to the weighted case, we first need a simple lemma:

Lemma 3 Let \( A \) be a symmetric nonnegative matrix and \( v \) be a positive vector, then the matrix \( \hat{A} = \text{diag} \left( \frac{Av}{v} \right) - A \) is positive semi-definite.

Proof. It is easy to see that \( \text{diag} \left( \frac{Av}{v} \right) = D_v^{-1}D_Av \), where \( D_x = \text{diag}(x) \) denote a diagonal matrix with the elements of the vector \( x \) as diagonal entries. The scaled version \( \hat{A}_s := D_v \hat{A}D_v \) of \( \hat{A} \) satisfies \( \hat{A}_s = D_AvD_v - D_vAD_v \) and is a diagonally dominant matrix since \( \hat{A}_s 1_m = (Av) \odot v - v \odot (Av) = 0 \) and its off-diagonal elements are negative. Therefore, the matrix \( \hat{A}_s \) is positive semi-definite, and so is \( \hat{A} \). □

We can now extend Theorem 1 to the weighted case.

Theorem 4 The weighted Euclidean distance \( \frac{1}{2} \|A - UV\|_W^2 \) is non-increasing under the updating rules:
\[ V \leftarrow V \odot \frac{[U^T (W \circ A)]}{[U^T (W \circ (UV))]}, \quad U \leftarrow U \odot \frac{[(W \circ A)V^T]}{[(W \circ (UV))V^T]}. \]  
(17)

The weighted Euclidean distance \( \frac{1}{2} \|A - UV\|_W^2 \) is invariant iff the conditions (10) and (12) hold.
Proof. We only treat the updating rule for $V$ since that of $U$ can be proven in a similar fashion. First, we point out that the cost $F(A, U V)$ splits in $n$ independent problems related to each column of the error matrix. We can therefore consider the partial cost function for a single column of $A$, $V$ and $W$, which we denote by $a$, $v$ and $w$, respectively:

$$F(v) = F_w(a, U v) = \frac{1}{2} \sum_i (w_i (a_i - [U v]_i)^2)$$

where $D_w = \text{diag}(w)$. Let $v^k$ be the current approximation of the minimizer of $F(v)$ then one can rewrite $F(v)$ as the following quadratic form:

$$F(v) = F(v^k) + (v - v^k)^T \nabla_v F(v^k) + \frac{1}{2} (v - v^k)^T D_w (a - U v)$$

where $\nabla_v F(v^k)$ is explicitly given by

$$\nabla_v F(v^k) = -U^T D_w (a - U v^k).$$

Next, we approximate $F(v)$ by a simpler quadratic model:

$$G(v, v^k) = F(v^k) + (v - v^k)^T \nabla_v F(v^k) + \frac{1}{2} (v - v^k)^T D(v^k) (v - v^k)$$

where $D(v^k)$ is a diagonal matrix chosen to make $D(v^k) - U^T D_w U$ positive semi-definite implying that $G(v, v^k) - F(v) \geq 0, \forall v$. The choice for $D(v^k)$ is similar to that proposed by Lee and Seung:

$$D(v^k) = \text{diag} \left( \frac{[U^T D_w U v^k]}{v^k} \right).$$

Lemma 3 assures the positive semi-definiteness of $D(v^k) - U^T D_w U$. As a result, we have

$$F(v^k) = G(v^k, v^k) \geq \min_v G(v, v^k) = G(v^{k+1}, v^k) \geq F(v^{k+1})$$

where $v^{k+1}$ is found by solving $\frac{\partial G(v, v^k)}{\partial v} = 0$:

$$v^{k+1} = v^k - D(v^k)^{-1} \nabla F(v^k)$$

$$= v^k + \text{diag} \left( \frac{v^k}{[U^T D_w U v^k]} \right) U^T D_w (a - U v^k)$$
Putting together the updating rules for all the columns of $V$ yields the desired result for the whole matrix $V$ in (17). The relation (24) shows that the weighted Euclidean distance is non-increasing under the updating rule for $V$, and (25) show that $v^{k+1} = v^k$ if and only if $v^k \circ \nabla F(v^k) = 0$. Finally, the non-negativity of $v^k$ is automatically satisfied.

4.2 The weighted generalized KL divergence

The following theorem generalizes Theorem 2 to the weighted case:

**Theorem 5** The weighted divergence $D_W(A \parallel UV)$ is non-increasing under the updating rules:

$$V \leftarrow \frac{[V]}{[U^T W]} \circ \left( U^T \frac{[W \circ A]}{[UV]} \right), \quad U \leftarrow \frac{[U]}{[W V^T]} \circ \left( \frac{[W \circ A]}{[UV]} V^T \right).$$

(30)

The weighted divergence $D_W(A \parallel UV)$ is invariant under these updates iff the conditions (10) and (12) hold.

**Proof.** Again, we prove the theorem only for $V$ and we also split the divergence into partial divergences corresponding to one column of $V$, $W$ and $A$, denoted by $v$, $w$ and $a$.

$$F(v) = D_w(a \parallel U v)$$

$$= \sum_i w_i \left( a_i \log a_i - a_i + \sum_j U_{ij} v_j - a_i \log \sum_j U_{ij} v_j \right).$$

(31)

This partial divergence is approximated by the following auxiliary function:

$$G(v, v^k) = \sum_i \left( w_i \left( a_i \log a_i - a_i + \sum_j U_{ij} v^k_j \right. \right.$$

$$\left. - a_i \sum_j \frac{U_{ij} v^k_j}{\sum_l U_{il} v^k_l} \left( \log U_{ij} v_j - \log \frac{U_{ij} v^k_j}{\sum_l U_{il} v^k_l} \right) \right)$$

(32)
Because of the convexity of the function $-\log(x)$ and since $\sum_j U_{ij}v^k_j = 1$, we have that $G(v, v^k) \geq F(v), \forall v$. Moreover $G(v^k, v^k) = F(v^k)$, so we obtain:

$$F(v^k) = G(v^k, v^k) \geq \min_v G(v, v^k) = G(v^{k+1}, v^k) \geq F(v^{k+1}).$$  

(33)

To obtain the updating rule, it is sufficient to construct the minimizer of $G$ with respect to $v$, given by:

$$\frac{\partial G(v, v^k)}{\partial v_j} = \sum_i w_i U_{ij} - v^k_j \sum_l U_{il}v^k_l = 0. \quad (34)$$

Then the minimizer of $G(v, v^k)$ is chosen as the next value of $v$:

$$v^{k+1} = \left[ \frac{v^k}{U^T w} \right] \circ \left( U^T \left[ a \circ w \right] [Uv^k] \right). \quad (35)$$

Putting together the updating rules for all the columns of $V$ gives the desired updating rule for the whole matrix $V$ as in (30). The relation (33) shows that the weighted divergence is non increasing under the updating rule for $V$. Using (35) and the fact that

$$\nabla F(v^k) = U^T w - U^T \left[ a \circ w \right] [Uv^k]$$  

(36)

we can easily see that that $v^{k+1} = v^k$ if and only if $v^k \circ \nabla F(v^k) = 0$. Finally, the non-negativity of $v^k$ is automatically satisfied. \hfill \Box

4.3 Linking the two cost functions

One can rewrite the updating rule for $V$ in the weighted generalized KL divergence case as follows:

$$V \leftarrow \frac{[V]}{[U^T W]} \circ \left( U^T \frac{[W \circ A]}{[U V]} \right) = V \circ \left( \frac{U^T \left[ W \circ (U^T A) \right]}{[U V]} \right) = V \circ \left( \frac{U^T (W_{UV} \circ A)}{U^T (W_{UV} \circ (U^T V))} \right), \quad (37)$$

where $W_{UV} = \frac{[W]}{[U V]}$. This shows that each update in the weighted generalized KL divergence is equivalent to an update in the weighted Euclidean distance with the weight matrix $W_{UV}$. This is an adaptive weighting since the weights
change after each update. And at the stationary point of this minimization, $V$ and $U$ converge to the minimizer of the weighted Euclidean distance for which the weight matrix is exactly $W_{UV}$.

Conversely, one can see that each update in the weighted Euclidean distance with the weight matrix $W$ is equivalent to an update in the weighted generalized KL divergence with the weight matrix $W_{UV} = W \circ (UV)$. And again, at the stationary point of this minimization, $U$ and $V$ converge to the minimizer of the weighted generalized KL divergence for which the weight matrix is exactly $W_{UV}$.

Moreover, if we look at the optimality conditions in the two cases

$$V \circ (U^T(W_1 \circ (UV - A))) = 0$$  \hspace{1cm} (38)

and

$$V \circ (U^T(W_2 \circ (1_{m \times n} - \frac{[A]}{[UV]}))) = 0,$$  \hspace{1cm} (39)

it is easy to see that if $W_1 = \frac{[W_2]}{[UV]}$, these two conditions are identical.

We summarize all the updating rules and the link between the two minimizations in the following table. In the unweighted case, the matrix $1_{m \times n}$ is included to make it easier to compare it with the matrices $W_1$ and $W_2$ of the weighted case. With our updating rules for the weighted case, we have thus shown that even though the two cost functions are very different, their minimizations are closely related.

<table>
<thead>
<tr>
<th>Table 1: Summary of algorithms for Weighted Nonnegative Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>NMF</td>
</tr>
<tr>
<td>WNMF</td>
</tr>
<tr>
<td>ED $\Leftrightarrow$ KLD</td>
</tr>
</tbody>
</table>

10
4.4 Other weighted NMF methods

The Nonnegative Matrix Factorization with Sparseness Constraint of [7] imposes sparseness constraints on the matrices $U$ and $V$. The algorithm uses two separate steps to achieve this: a gradient-descent step and a sparseness control step. Weights can be easily added in the gradient-descent step by setting the cost function to be the weighted Euclidean distance instead of the unweighted one. The sparseness control step is kept unchanged.

For other NMMF methods like Fisher Nonnegative Matrix Factorization [14], Localized Nonnegative Matrix Factorization [10] etc., weighted version of iterative algorithms can also easily be obtained.

5 Face Feature Extraction by NMF

In [8] Lee and Seung argued that there is a link between human perception and nonnegative data representation. The intuition behind this is that perception is based on a representation that is additive and tends to expose parts of the data. Since then, many researchers have tried to use nonnegative representations of data – such as NMF – in many application areas.

One of the major application of NMF is the representation of human faces. In this section, we show the results of two numerical experiments on human faces. These experiments also illustrate the effect of weights on the obtained approximation.

5.1 Experiment settings

The experiments use the Cambridge ORL face database as the input data. The database contains 400 images of 40 persons (10 images per person). The size of each image is $112 \times 92$ with 256 gray levels per pixel representing a front view of the face of a person. As was also done in earlier papers, we chose here to show the images in negative because visibility is better. Pixels with higher intensity are therefore darker. Ten randomly chosen images are shown in the first row of Figure 1.

The images are then transformed into 400 “face vectors” in $\mathbb{R}^{10304}$ ($112 \times 92 = 10304$) to form the data matrix $A$ of size $10304 \times 400$. We used three weight matrices of the same size of A (ie. $10304 \times 400$).

- **Uniform weight** $W_1$: a matrix with all elements equal to 1 (i.e. the unweighted case).
- **Image-centered weight** $W_2$: a nonnegative matrix whose columns are identical, i.e. the same weights are applied to every images. For each image,
Fig. 1. Original faces (first row), their image-centered weights $W_2$ (second row) and their face-centered weights $W_3$ (last row).

The weight of each pixel is given by $w_d = e^{-\frac{d^2}{\sigma^2}}$ where $\sigma = 30$ and $d$ is the distance of the pixel to the center of the image (56.5, 46.5). This weight matrix has rank one. Ten columns of this matrix are shown in the second row of Figure 1.

- **Face-centered weight $W_3$:** a nonnegative matrix whose columns are not identical, i.e. different weights are applied to different images. For each image, the weight of each pixel is given by $w_d = e^{-\frac{d^2}{\sigma^2}}$ where $\sigma = 30$ and $d$ is the distance of the pixel to the center of the face in that image. The rank of this matrix is not restricted to one. Ten columns of this matrix are shown in the last row of Figure 1.

Next, the matrix $A$ is approximated by nonnegative matrices $U$ and $V$. The rank chosen for the factorization is 49, the matrices $U$ and $V$ will thus be of dimension $10304 \times 49$ and $49 \times 400$ respectively. Each column of $U$ is considered as a nonnegative basis vector. And the storing space for the approximation will be $10304 \times 49 + 49 \times 400$ which is much smaller than $10304 \times 400$ for the data matrix $A$.

### 5.2 NMF versus Weighted NMF

In this experiment, all three weight matrices $W_1$, $W_2$ and $W_3$ are used in a NMF based on the weighted generalized KL divergence. For each weight matrix, 49 nonnegative bases, i.e. columns of $U$, are calculated and shown in Figure 2.

Each image in the database can be reconstructed as a weighted sum of these nonnegative bases with nonnegative weights determined by the corresponding column of $V$. In Figure 3, ten selected images are compared with the reconstructed images from the three experiments. The pixel-wise generalized KL divergence averages from the three experiments are shown in Figure 4.
It can be seen from the results that more important pixels (i.e. those with higher weight, at the center of images or at the center of faces in our example) are better reconstructed than less important ones. This improvement can be seen in both reconstructed images and the pixel-wise average divergence of all the images. In figure 4, all the images are shifted to have a common face center. The darker colors correspond to larger errors, which means that the algorithm pays more attention to the center of the images (or to the center of the faces) and that the details at the center areas are privileged in the approximation. More details can be seen on the reconstructed faces when face-centered weights are applied, especially when the center of a face is further away from the center of the image. And for each of the three cases (unweighted, image-centered and face-centered), the approximation errors in KL divergence of different weight
Fig. 4. Pixel-wise average divergence: unweighted (left), image-centered (middle) and face-centered (right)

are shown in Table 2.

<table>
<thead>
<tr>
<th>Weight Type</th>
<th>$D_{W_1}(A∥U_∞V_∞)$</th>
<th>$D_{W_2}(A∥U_∞V_∞)$</th>
<th>$D_{W_3}(A∥U_∞V_∞)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniform weight</td>
<td>925.6524</td>
<td>112.8759</td>
<td>121.0499</td>
</tr>
<tr>
<td>Image-centered weight</td>
<td>1142.4479</td>
<td>81.9766</td>
<td>94.2886</td>
</tr>
<tr>
<td>Face-centered weight</td>
<td>1243.3588</td>
<td>87.8281</td>
<td>83.9154</td>
</tr>
</tbody>
</table>

The results for weight matrix $W_3$ also show that our algorithms can deal with weight matrices without rank restriction. And weights can be adapted to each data vector in order to yield better approximations.

5.3 NMF with Sparseness Constraint versus Weighted NMF with Sparseness Constraint

This second experiment shows the effect of adding weights into the NMF with Sparseness Constraint. Figure 5 shows two sets of 49 nonnegative bases obtained by the NMF with Sparseness Constraint with uniform weight $W_1$ (left) and with face-centered weight $W_3$ (right).

The NMF with Sparseness Constraint is often used to extract local and independent features on faces. As weights are more centered, more features at the center of faces are retained as we can see in Figure 6. This allows us to tune the NMF with Sparseness Constraint algorithm to more relevant parts to give more useful information about the data.
6 Conclusion

In this paper, we extended some Nonnegative Matrix Factorization (NMF) algorithms in order to incorporate weighting matrices and we derived weighted iterative schemes for which we proved convergence results that are similar to the unweighted counterparts. We showed that the inclusion of weights allowed us to link the different algorithms in a certain manner and we showed that weighting yields an important flexibility allowing to better emphasize certain features in image approximation problems. This was illustrated in the approximation of faces extracted from a database that is often used as benchmark.
Acknowledgements

This paper presents research supported by the Concerted Research Action (ARC) ”Large Graphs and Networks” of the French Community of Belgium, and by the Belgian Programme on Inter-university Poles of Attraction, initiated by the Belgian State, Prime Minister’s Office for Science, Technology and Culture. The scientific responsibility rests with the authors. Ngoc-Diep Ho is a FRIA fellow.

References


