NONNEGATIVE MATRIX FACTORIZATION WITHOUT NONNEGATIVITY CONSTRAINTS ON THE FACTORS

BART VANLUYTEN*, JAN C. WILLEMS†, AND BART DE MOOR‡

Abstract. We consider a new kind of low rank matrix approximation problem for nonnegative matrices: given a nonnegative matrix M, approximate it with a low rank product $V.H$ such that $V.H$ is nonnegative, but without nonnegativity constraints on V and H separately. The nonnegativity constraint on $V.H$ is natural when using the Kullback-Leibler divergence as optimality criterion. We propose an iterative algorithm that locally minimizes the Kullback-Leibler divergence between M and $V.H$. In addition, we show that the method can be easily adapted to impose an upper- and lower bound on the elements of $V.H$. Finally, we apply the algorithm to a data compression problem for a database containing human faces and compare it with other existing methods.

Key words. Matrix factorization, Kullback-Leibler divergence, nonnegative matrices, iterative method

AMS subject classifications. 15A23, 15A48

1. Introduction. Nonnegative matrices play an important role in engineering applications as many variables are constrained to be nonnegative from physical point of view (greyscale intensity values, concentrations, financial data,...). It is therefore of no surprise that during recent years the Nonnegative Matrix Factorization (NMF) [7] gained in interest, both from theoretical point of view [2, 4] as well as in applications [1, 9].

The general low rank matrix approximation problem of an $m_1 \times m_2$ real matrix M with a product $V.H$ with inner dimesion a ($a < m_1, m_2$) consists of minimizing $D(M, V.H)$ over V and H, where $D(X, Y)$ is a distance measure between X and Y. If there are no nonnegativity constraints on the factors V and H, the Singular Value Decomposition (SVD) gives a decomposition that is optimal in the Frobenius distance. There exist efficient algorithms to compute the SVD [5].

In some applications it is required to impose nonnegativity constraints on the factors V and H. By imposing such constraints the problem becomes difficult to solve. However, one can still look for a local minimum of $D(M, V.H)$ for certain choices of the distance measure $D(X, Y)$. Iterative algorithms that converge to such local minima have been proposed in [7, 8].

However, in certain applications, the nonnegativity of V and H separately is not important, but only the nonnegativity of the approximation $V.H$ matters. In this paper, we consider the low rank matrix approximation problem with the constraint that $V.H$ is nonnegative but without nonnegativity constraints on V and H separately. Such a low rank approximation can still be interpreted from physical point of view, and the low rank factorization allows a data reduction. We show that the nonnegativity constraint on $V.H$ is automatically satisfied by using the Kullback-Leibler divergence as distance measure between M and $V.H$ and we present an iterative algorithm that converges to a local optimum of the divergence. We refer to this decomposition as the Matrix Factorization (MF) in contrast to the Nonnegative Matrix Factorization (NMF).

The NMF can be used for any matrix factorization problem where the elements of the product $V.H$ are bounded below by an arbitrary number (not necessary 0). The NMF problem

*Electrical Engineering Department, K.U.Leuven, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium (bart.vanluyten@esat.kuleuven.be).

†Electrical Engineering Department, K.U.Leuven, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium (jan.willems@esat.kuleuven.be).

‡Electrical Engineering Department, K.U.Leuven, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium (bart.demoor@esat.kuleuven.be).

does not consider upper bounds on the elements of $V.H$. In many applications, upper bounds are just as important as lower bounds, however. We show that our decomposition can easily be adapted to impose both lower and upper bounds on the elements of the product $V.H$.

In Section 2, we give a motivation for the matrix factorization approach. In Section 3, we review the classical nonnegative matrix factorization problem, and in Section 4 we formally introduce the new matrix factorization problem and prove some interesting properties of the decomposition. In Section 5, we give an algorithm to compute the matrix factorization and in Section 6 we explain how the algorithm can be adapted to deal with upper bounds. In Section 7, we apply the algorithm to the data compression problem on a database of human faces.

The following notation is used. $\mathbb{R}_+$ is the set of nonnegative real numbers. If X is a matrix, then X_{ij} denotes its i,j-th element, $X_{i,:}$ denotes the i-th row and $X_{i:j,:}$ the matrix formed by the i-th to the j-th row of X. Analogous notations are used for selecting columns instead of rows. $X \geq 0$ means that the elements of X are nonnegative. With e we denote the column vector with all elements equal to 1, that is $e = [\,1\ 1\ \ldots\ 1]^T$, and with $\mathbf{1}$, we denote a matrix with all elements equal to 1. The dimensions of e and $\mathbf{1}$ are clear from the context.

2. Intuitive motivation. In the paper of Lee and Sueng [7] the nonnegative matrix factorization is used to model a database of facial images. Each face in the database is represented by a nonnegative column vector, leading to a nonnegative matrix with the number of columns equal to the number of faces in the database. Next the matrix is factored as the product $V.H$ with the matrices V and H nonnegative. The columns of V contain *eigenfaces* and the columns of H are nonnegative weights that reconstruct the original faces as linear combinations of the eigenfaces. By carrying out an approximation of M as $V.H$ with V and H nonnegative, an efficient compression of the original database can be obtained. A first argument for using the nonnegative matrix factorization instead of the SVD or other matrix decompositions is that the reconstructed images $V.H$ contain only nonnegative elements and are therefore interpretable as facial images. A second argument is that the eigenfaces are typically sparse (i.e. many zero elements) because of their nonnegativity, which makes that there is an *part-based* interpretation of decomposition, i.e. each face is decomposed into sum of a small number of *parts* (nose, eyes, ears, ...). In our opinion, the first argument is always relevant, while the second is not important in certain applications (for instance in the application where the decomposition is only used for data compression). We therefore consider the decomposition of a nonnegative matrix M into a nonnegative product $V.H$, but where the factors V and H themselves are allowed to contain negative values. In this section, we give an intuitive example motivating our approach.

Suppose one wants to model moustaches and beards of a database of human faces with 4 parameters. The first parameter represents the top of the moustache (closest to the nose) and the second for the bottom of the moustache. The third parameters is used for the top of the beard (closest to the mouth) and the fourth for the bottom of the beard (see figure 2.1(a)). The values for the parameters range from 0 (no moustache/beard) over 0.5 (light-coloured moustache/beard) to 1 (dark moustache/beard). In figure 2.1 (b)-(e), we show faces with four different possible combinations of dark beards and moustaches.

The matrix M, which contains in its i-th column the 4 parameters of face i is given by

$$M = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}.$$

This matrix M has rank 3, but positive rank 4 (see [10]). This means that there exists an exact decomposition of inner dimension 3 if negative factors V and H are allowed, but that

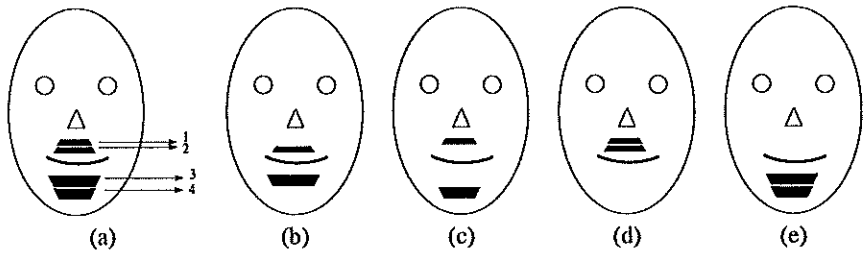


FIG. 2.1. *(a) Interpretation of the 4 parameters used to model human moustaches and beards, (b)-(e) Four different examples of combinations of beard and moustaches.*

the smallest inner dimension for a decomposition with nonnegative factors is equal to 4, i.e.

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & -1 \\ -1 & 1 & 0 & 0 \end{bmatrix}$$
$$= \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Of course, one can also build an approximate decomposition of inner dimension 3 with nonnegative factors,

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \simeq \begin{bmatrix} 0 & \frac{1}{3} & \frac{1}{2} \\ 0 & \frac{2}{3} & 0 \\ \frac{2}{3} & 0 & 0 \\ \frac{1}{3} & 0 & \frac{1}{2} \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 2 \\ 1 & 0 & 2 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{3} & 1 & \frac{2}{3} & 0 \\ \frac{2}{3} & 0 & \frac{4}{3} & 0 \\ \frac{2}{3} & 0 & 0 & \frac{4}{3} \\ \frac{1}{3} & 1 & 0 & \frac{2}{3} \end{bmatrix}.$$

but in that case the decomposition is not exact anymore. With this academic example, we show that by allowing negative elements in the factors of the decomposition, we can find an exact decomposition of lower rank than the decomposition that is found when only nonnegative factors are allowed. Here it is clear that the decomposition with negative factors gives rise to a nonnegative product $V.H$ as the decomposition is exact.

In this paper, we propose a decomposition for which $V.H$ is nonnegative (but V and H not) even if the decomposition is not exact. So our decomposition finds an approximation of the same quality as the NMF with a smaller inner dimension (so a larger data reduction) or finds a better approximation if the same inner dimension is used as for the NMF.

In section 7, we illustrate that the effects that we have seen in this intuitive example extend to larger matrix decomposition problems.

3. Nonnegative matrix factorization. The NMF problem can be stated as follows: Given a matrix $M \in \mathbb{R}_+^{m_1 \times m_2}$, find a decomposition $M = V.H$ with $V \in \mathbb{R}_+^{m_1 \times a}$ and $H \in \mathbb{R}_+^{a \times m_2}$, and with a as small as possible. The minimal inner dimension a for which such a decomposition exists is called the *positive rank* (p–rank) of M. It is clear that $0 \leq \text{rank}(V) \leq \text{p–rank}(V) \leq \min\{m_1, m_2\}$. There do not exist effective algorithms to find the positive rank of a general positive matrix. Furthermore, no exact algorithm is

known to compute a nonnegative matrix factorization. Recently, the approximate nonnegative matrix factorization problem has been introduced in [7]. The idea is to choose the inner dimension a first and determine matrices V and H such that $V.H$ approximates M optimally in a certain distance measure. The Kullback-Leibler divergence is a popular such measure. The Kullback-Leibler divergence between two nonnegative matrices of the same size is defined as

$$D_{\mathrm{KL}}(X||Y) = \sum_{ij}(X_{ij}\log\frac{X_{ij}}{Y_{ij}} - X_{ij} + Y_{ij}).$$

The approximate nonnegative matrix factorization problem can now be stated as

PROBLEM 1. *Given $M \in \mathbb{R}_+^{m_1\times m_2}$ and $a \in \mathbb{N}$, minimize $D_{\mathrm{KL}}(M||V.H)$ with respect to V ($\in \mathbb{R}^{m_1\times a}$) and H ($\in \mathbb{R}^{a\times m_2}$), subject to the constraints $V, H \geq 0$.*

This minimization problem is convex in V and H separately, but is not convex in V and H jointly. Hence, it may have many local minima. Lee and Sueng [8] propose the following algorithm to solve Problem 1.

ALGORITHM 1. *Set V equal to an arbitrary matrix in $\mathbb{R}_+^{m_1\times a}$ and H to an arbitrary matrix in $\mathbb{R}_+^{a\times m_2}$. Now execute the following algorithm.*

until convergence do

$$H_{i,l} \leftarrow H_{i,l}\frac{\sum_\mu V_{\mu i}\frac{M_{\mu l}}{(V.H)_{\mu l}}}{\sum_\mu V_{\mu i}}$$

$$V_{k,i} \leftarrow V_{k,i}\frac{\sum_\nu H_{i\nu}\frac{M_{k\nu}}{(V.H)_{k\nu}}}{\sum_\nu H_{i\nu}}$$

end until

The algorithm is motivated by the following theorem.

THEOREM 1. *[7, 8] The divergence $D_{\mathrm{KL}}(M||V.H)$ is nonincreasing under the update rules*

$$H_{i,l} \leftarrow H_{i,l}\frac{\sum_\mu V_{\mu i}\frac{M_{\mu l}}{(V.H)_{\mu l}}}{\sum_\mu V_{\mu i}}, \quad V_{k,i} \leftarrow V_{k,i}\frac{\sum_\nu H_{i\nu}\frac{M_{k\nu}}{(V.H)_{k\nu}}}{\sum_\nu H_{i\nu}}.$$

The divergence is invariant under these updates if and only if (V, H) is a stationary point of the divergence.

If the initial values for V and H are chosen to be nonnegative as we assume, then the matrices V and H obtained are also nonnegative.

4. Nonnegative matrix factorization without nonnegativity constraints on the factors. The approximate nonnegative matrix factorization without nonnegativity constraints on the factors can be stated as

PROBLEM 2. *Given $M \in \mathbb{R}_+^{m_1\times m_2}$ and $a \in \mathbb{N}$, minimize $D_{\mathrm{KL}}(M||V.H)$ with respect to V ($\in \mathbb{R}^{m_1\times a}$) and H ($\in \mathbb{R}^{a\times m_2}$), subject to the constraint $V.H \geq 0$.*

Notice that the solution of Problem 2 is not unique. If (V, H) is a solution the problem, then $(VT, T^{-1}H)$, with T a nonsingular matrix in $\mathbb{R}^{a\times a}$ is also a solution. Next, it turns out that the constraint $V.H \geq 0$ is automatically fullfilled by using the Kullback-Leibler divergence as distance criterion. Indeed, the Kullback-Leibler divergence $D_{\mathrm{KL}}(M||V.H)$ goes to infinity if an element of $V.H$ goes to 0, while the corresponding element of M is not equal to 0, and the divergence is undefined if an element of $V.H$ is negative.

The partial derivatives of $F(V, H) = D_{\mathrm{KL}}(M||V.H)$ with respect to the elements V_{ki} and H_{il} are

$$\frac{\partial F}{\partial V_{ki}}(V,H) = -\sum_{\nu} H_{i\nu}\frac{M_{k\nu}}{(V.H)_{k\nu}} + \sum_{\nu} H_{i\nu}, \tag{4.1}$$

$$\frac{\partial F}{\partial H_{il}}(V,H) = -\sum_{\mu} V_{\mu i}\frac{M_{\mu l}}{(V.H)_{\mu l}} + \sum_{\mu} V_{\mu i}, \tag{4.2}$$

Hence the conditions for stationarity of (V, H) with respect to $D_{\mathrm{KL}}(M||V.H)$ are

$$\frac{\partial F}{\partial V_{ki}}(V,H) = 0, \qquad \frac{\partial F}{\partial H_{il}}(V,H) = 0, \tag{4.3}$$

for $i = 1, 2, \ldots, a$, $k = 1, 2, \ldots, m_1$ and $l = 1, 2, \ldots, m_2$.

In the case $a = 1$, Problem 2 can be solved explicitely. Moreover, the global optimizing solution (V_1, H_1) is equal to the optimal solution of Problem 1 and is given by

$$(V_1)_{k,1} = \frac{Me}{\sqrt{e^{\top}Me}}, \qquad (H_1)_{1,l} = \frac{e^{\top}M}{\sqrt{e^{\top}Me}}. \tag{4.4}$$

For the case $a > 1$, we propose in Section 5 an iterative algorithm that converges to a stationary point of F. In the remainder of this section, we establish some interesting properties of the stationary points of Problem 2. The first property is proven following the approach of [6].

PROPOSITION 1. *Let $M \in \mathbb{R}^{m_1 \times m_2}$. Every stationary point (V, H) of the cost function $D_{\mathrm{KL}}(M||V.H)$ preserves the row and column sum of M, i.e.*

$$Me = (V.H)e \tag{4.5}$$

$$e^{\top}M = e^{\top}(V.H) \tag{4.6}$$

As a consequence the element sum of M is preserved, i.e. $e^{\top}Me = e^{\top}(V.H)e$.

Proof: Equation (4.3) gives, for $k = 1, 2, \ldots, m_1$ and $i = 1, 2, \ldots, a$,

$$\sum_{\nu} H_{i\nu}\frac{M_{k\nu}}{(V.H)_{k\nu}} = \sum_{\nu} H_{i\nu}.$$

Multiplying with V_{ki} and summing over i proves equation (4.5). The proof of equation (4.6) is analogous.

■

We now prove a second property of stationary points.

PROPOSITION 2. *Let $M \in \mathbb{R}^{m_1 \times m_2}$. Given a stationary point (V, H) of the cost function $D_{\mathrm{KL}}(M||V.H)$, there exists a nonsingular matrix $T \in \mathbb{R}^{a \times a}$ such that $\tilde{V} = VT$ and $\tilde{H} = T^{-1}H$ satisfy the following conditions*

$$\tilde{V}(:,1) = \frac{Me}{\sqrt{e^{\top}Me}}, \tag{4.7}$$

$$\tilde{H}(1,:) = \frac{e^{\top}M}{\sqrt{e^{\top}Me}}. \tag{4.8}$$

Proof: Define

$$V' = VR$$

$$H' = R^{-1}H$$

with $R = \begin{bmatrix} \frac{He}{\sqrt{e^\top Me}} & R' \end{bmatrix}$ where R' is such that R is nonsingular. Since $He \neq 0$ such a matrix R' always exists. Then

$$V'(:,1) = V\frac{He}{\sqrt{e^\top Me}} = \frac{Me}{\sqrt{e^\top Me}}$$

Calculate now

$$\tilde{V} = V'S^{-1}$$
$$\tilde{H} = SH'$$

with $S = \begin{bmatrix} \frac{e^\top V'}{\sqrt{e^\top Me}} \\ 0 \quad S' \end{bmatrix}$, where S' is such that S is nonsingular. Since $\frac{e^\top V'}{\sqrt{e^\top Me}}(1) = 1$, such a matrix S' always exists. Hence

$$\tilde{H}(1,:) = \frac{e^\top V'}{\sqrt{e^\top Me}}H' = \frac{e^\top M}{\sqrt{e^\top Me}}.$$

From $S_{1,1} = \frac{\sqrt{e^\top Me}}{\sqrt{e^\top Me}} = 1$, we obtain that the first column of S^{-1} is equal to $[\,1\,0\,0\,\ldots\,0]^\top$, and hence

$$\tilde{V}(:,1) = \frac{Me}{\sqrt{e^\top Me}}.$$

■

A third property of the matrix factorization is related to the nesting property of the singular value decomposition (SVD). The nesting property of the SVD says that the optimal rank b (with $b < a$) approximation of the optimal rank a approximation of a given matrix M is equal to the optimal rank b approximation of the original matrix M. In addition the optimal rank a SVD-approximation can be written in a form $M_a = \tilde{V}_a\tilde{H}_a$ such that the optimal rank b approximation is equal to $M_b = V_bH_b$, where $V_b = (\tilde{V}_a)_{:,1:b}$ and $H_b = (\tilde{H}_a)_{1:b,:}$. For the matrix factorization of this paper, we have the same properties, but only for $b = 1$. Indeed, from Proposition 1 and equation (4.4), we know that the optimal rank 1 approximation $M_1 = V_1H_1$ of an optimal rank a approximation V_aH_a of a matrix M is equal to the optimal rank 1 approximation of the original matrix M. In addition, proposition 2 says that the optimal approximation of rank a can be written in a form $M_a = \tilde{V}_a\tilde{H}_a$ such that $(\tilde{V}_a)_{:,1} = V_1$ and $(\tilde{H}_a)_{1,:} = H_1$.

5. Algorithm. Analogously to the nonnegative matrix factorization problem, the matrix factorization problem (Problem 2) is convex in V and H separately, but is not convex in V and H jointly. Therefore, we use an approach where V and H are updated alternatingly. In the proposed algorithm, these updates are performed using one step of the damped Newton method. In a step of the damped Newton method, one first builds a quadratic approximation of the cost function in the current value for the optimization variable. Next, one makes a step in the direction of the minimum of this quadratic function. The step size is determined by a backtracking approach. Formally, a Newton step to optimize the function $f : \mathbb{R}^n \mapsto \mathbb{R}$ is given by

$$x \leftarrow x - \mu\left[\frac{\partial^2 f}{\partial x_i \partial x_j}(x)\right]^{-1}\left[\frac{\partial f}{\partial x_i}(x)\right],$$

where $\left[\frac{\partial f}{\partial x_i}\right]$ is the gradient and $\left[\frac{\partial^2 f}{\partial x_i \partial x_j}\right]$ is the Hessian of f, and μ is the stepsize.

We first discuss the updating of H. The optimization problem with cost $F(V,H) = D_{\mathrm{KL}}(M||V.H)$ splits up into m_2 independent problems related to each column of H. We therefore consider the partial cost for a single column of M and H which we denote by $M_{:,l}$ and $H_{:,l}$. The gradient $\in \mathbb{R}^{1\times a}$ of the cost function $F(V,H_{:,l}) = D_{\mathrm{KL}}(M_{:,l}||V.H_{:,l})$ is given by (4.2) and the Hessian $\in \mathbb{R}^{a\times a}$ is given by

$$\frac{\partial^2 F}{\partial H_{il}\partial H_{jl}}(V,H) = \sum_{\mu} V_{\mu i}V_{\mu j}\frac{M_{\mu l}}{((V.H)_{\mu l})^2}.$$

The update rule for $H_{:,l}$ becomes

$$H_{:,l} \leftarrow H_{:,l} - \left[\frac{\partial^2 F}{\partial H_{il}\partial H_{jl}}(V,H)\right]^{-1}\left[\frac{\partial F}{\partial H_{il}}(V,H)\right].$$

The updating rule for V is completely analogous. The Hessian of the cost function $F(V_{k,:},H) = D_{\mathrm{KL}}(M_{k,:}||V_{k,:}H)$ is given by

$$\frac{\partial^2 F}{\partial V_{ki}\partial V_{kj}}(V,H) = \sum_{\nu} H_{i\nu}H_{j\nu}\frac{M_{k\nu}}{((V.H)_{k\nu})^2}.$$

A formal statement of the complete algorithm is given in Algorithm 2. The parameter α can be chosen by the user. A typical value is $\alpha = 0.5$.

ALGORITHM 2. *Set $\alpha \in [0,1]$, V to an arbitrary matrix in $\mathbb{R}_+^{m_1\times a}$ and H to an arbitrary matrix in $\mathbb{R}_+^{a\times m_2}$. Now execute the following algorithm.*

until** convergence **do
  $\mu = 1$
  until** $V.H$ is nonnegative **do
    $H_{:,l}^{\mathrm{new}} \leftarrow H_{:,l} - \mu\left[\frac{\partial^2 F}{\partial H_{il}\partial H_{jl}}(V,H)\right]^{-1}\left[\frac{\partial F}{\partial H_{il}}(V,H)\right], \quad (l = 1,2,\ldots m_2)$
    $\mu = \alpha\mu$
  end until
  $H \leftarrow H^{\mathrm{new}}$
  $\nu = 1$
  until** $V.H$ is nonnegative **do
    $V_{k,:}^{\mathrm{new}} \leftarrow V_{k,:} - \nu\left[\frac{\partial^2 F}{\partial V_{ki}\partial V_{kj}}(V,H)\right]^{-1}\left[\frac{\partial F}{\partial V_{ki}}(V,H)\right], \quad (k = 1,2,\ldots m_1)$
    $\nu = \alpha\nu$
  end until
  $V \leftarrow V^{\mathrm{new}}$
end until

In the algorithm the inverse of the Hessian has to be computed. This computation can give rise to two different kind of problems. We describe both problems and shortly discuss a possible solution. The first possible problem is that the Hessian can become singular such that its inverse can not be computed. A solution therefore is to add λI to the Hessian before inverting it, where λ is a to be chosen constant and I is the unit matrix of appropriate size. This approach is known as the Levenberg-Marquardt method. A second problem is that the computation and storage of the Hessian becomes very expensive when decomposing a large-scale matrix. A solution therefore is to use other optimization methods that do not need the Hessian for computing an update, such as gradient or conjugate gradient methods.

The observation that the optimal rank a decomposition of M can be transformed to a form such that it *contains* the optimal rank one decomposition (see Section 4), can be used to simplify the algorithm to find the decomposition. Indeed, one can fix the first column of V_a to $Me/\sqrt{e^\top Me}$ and the first row of H_a to $e^\top M/\sqrt{e^\top Me}$, and optimize over the second to the last columm (row) of V_a (H_a).

In some applications, it is necessary to impose that only V is nonnegative as well as the product $V.H$, but that the nonnegativity of H is not important. In that case, one can easily combine the update formula for V of the nonnegative matrix factorization with the update formulas for H of the matrix factorization. A similar construction holds, of course, if H and $V.H$ are required to be nonnegative, but not V.

6. Imposing upper bounds. The nonnegative matrix facorization has been used in many applications where the variables are constrained to be nonnegative from physical point of view. However, in some applications, the variables are not only bounded from below, but are also bounded from above. For instance, the greyscale value of an image is usually a number between 0 and 1. Therefore in this application, it is odd to impose the lower bound 0, while not imposing the upper bound 1. The algorithm given in Section 5 can easily deal with upper bounds. Introduce a modified version of the Kullback-Leibler divergence

$$\begin{aligned} D_{\mathrm{MKL}}(X||Y) &= D_{\mathrm{KL}}(X||Y) + D_{\mathrm{KL}}(1-X||1-Y) \\ &= \sum_{ij} X_{ij} \log \frac{X_{ij}}{Y_{ij}} + \sum_{ij} (1-X_{ij}) \log \frac{1-X_{ij}}{1-Y_{ij}}. \end{aligned}$$

Now consider the following problem

PROBLEM 3. *Given* $M \in \mathbb{R}_+^{m_1 \times m_2}$ *and* $a \in \mathbb{N}$, *minimize* $G(V,H) = D_{\mathrm{MKL}}(M||V.H)$ *with respect to* V ($\in \mathbb{R}^{m_1 \times a}$) *and* H ($\in \mathbb{R}^{a \times m_2}$), *subject to the constraints* $0 \leq (V.H)_{kl} \leq 1$.

The constraints $0 \leq (V.H)_{kl} \leq 1$ are automatically fullfilled by using the modified version of the Kullback-Leibler divergence. Indeed, the modified Kullback-Leibler divergence is undefined if $V.H$ is negative or larger than 1, and in addition, the modified divergence $D_{\mathrm{MKL}}(M||V.H)$ goes to infinity if an element of $V.H$ goes to 0 or to 1, while the corresponding element of M is not equal to 0 or 1.

Our approximate matrix factorization algorithm can be easily adapted to deal with Problem 3 by using the followings expressions for the gradient and Hessian.

$$\begin{aligned} \frac{\partial G}{\partial V_{ki}}(V,H) &= -\sum_{\nu} H_{i\nu} \frac{M_{k\nu}}{(V.H)_{k\nu}} + H_{i\nu} \frac{1-M_{k\nu}}{1-(V.H)_{k\nu}}, \\ \frac{\partial G}{\partial H_{il}}(V,H) &= -\sum_{\mu} V_{\mu i} \frac{M_{\mu l}}{(V.H)_{\mu l}} + V_{\mu i} \frac{1-M_{\mu l}}{1-(V.H)_{\mu l}}, \\ \frac{\partial^2 G}{\partial V_{ki} \partial V_{kj}}(V,H) &= \sum_{\nu} H_{i\nu} H_{j\nu} \frac{M_{k\nu}}{((V.H)_{k\nu})^2} + H_{i\nu} H_{j\nu} \frac{1-M_{k\nu}}{(1-(V.H)_{k\nu})^2}, \\ \frac{\partial^2 G}{\partial H_{il} \partial H_{jl}}(V,H) &= \sum_{\mu} V_{\mu i} V_{\mu j} \frac{M_{\mu l}}{((V.H)_{\mu l})^2} + V_{\mu i} V_{\mu j} \frac{1-M_{\mu l}}{(1-(V.H)_{\mu l})^2}. \end{aligned}$$

7. Application. In this application, we use data from the CBCL-database of human faces [3]. This database contains 2429 facial greyscale images of size 19×19. We use the first 100 images, vectorize and stack them in the columns of the matrix M as is done in [7]. We then approximate the matrix M with a product $V.H$ with inner dimension $a = 20$ with four different procedures: nonnegative matrix factorization (NMF), the singular value

TABLE 7.1
Kullback-Leibler divergence and modified Kullback-Leibler divergence between M and the decomposition $V.H$ for the 4 different decomposition methods.

	MF	UBMF	NMF	SVD
KL-divergence	339	383	564	460 (*)
MKL-divergence	1104 (*)	806	1470 (*)	989 (*)

decomposition (SVD), the matrix factorization of this paper (MF), and the upper bounded matrix factorization of this paper with a lower (0) and upper bound (1) on the elements of $V.H$ (UBMF). The NMF, MF, UBMF algorithms converged using less than 50 iteration steps, but the results shown are after 100 iterations.

Table 7.1 shows that MF and UBMF give the best results, illustrated by the Kullback-Leibler and the modified Kullback-Leibler divergence between M and the approximation $V.H$ computed with the four different methods. For all methods, pixel values of the reconstruction that are ≤ 0 or ≥ 1, are truncated to the values 0.0001 and 0.9999 respectively. We do not truncate these values to 0 and 1, as this operation makes the distances indicated with (*) equal to infinity which makes comparison between the methods impossible.

In figure 7.1, we plot face 34 and 70 and their reconstructions using the 4 different methods. Pixel values smaller than 0 are plotted in red and pixel values higher than 1 are plotted in blue. From this plot, we conclude that the matrix factorization (MF) and upper bounded matrix factorization (UBMF) give the best results.

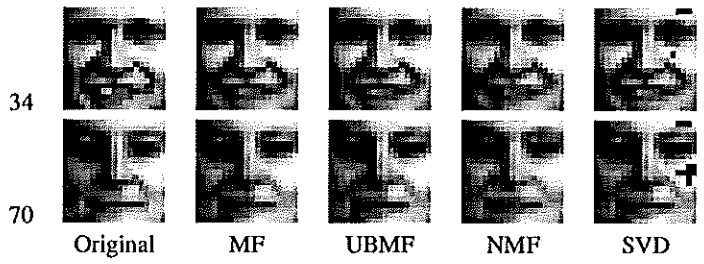


FIG. 7.1. *Plot of the i-th face of the database, with $i = 34, 70$, and their reconstructions using the matrix factorization of this paper (MF), the matrix factorization with upper bounds (UBMF), the nonnegative matrix factorization (NMF) and the singular value decomposition (SVD). Pixel values smaller than 0 are plotted in red and pixel values higher than 1 are plotted in blue.*

8. Conclusions. In this paper, we considered a matrix approximation problem for nonnegative matrices: Given a nonnegative matrix M, approximate it with a low rank product $V.H$ such that $V.H$ is nonnegative, but without nonnegativity constraints on V and H separately. We show that the nonnegativity constraint on $V.H$ is natural when using the Kullback-Leibler divergence as optimality criterion and proved some properties of the matrix factorization. We propose an iterative algorithm that locally minimizes that Kullback-Leibler divergence between M and $V.H$. In addition, we show that the method can be easily adapted to impose an upper- and a lower bound on the elements of $V.H$ instead of only a lower bound. Finally, we applied the algorithm to a data compression problem on a database containing human faces and explained the advantages compared to existing methods.

Acknowledgements. Bart Vanluyten is a research assistant with the Fund for Scientific Research Flanders (FWO-Vlaanderen). Jan Willems and Bart De Moor are professor with the Katholieke Universiteit Leuven. The SISTA research program is supported by: Research Council KUL: GOA AMBioRICS, CoE EF/05/006 Optimization in Engineering, several PhD/postdoc and fellow grants; Flemish Government: FWO: PhD/postdoc grants,

projects, G.0407.02 (support vector machines), G.0197.02 (power islands), G.0141.03 (Identification and cryptography), G.0491.03 (control for intensive care glycemia), G.0120.03 (QIT), G.0452.04 (new quantum algorithms), G.0499.04 (Statistics), G.0211.05 (Nonlinear), G.0226.06 (cooperative systems and optimization), G.0321.06 (Tensors), G.0302.07 (SVM/Kernel, research communities (ICCoS, ANMMM, MLDM); IWT: PhD Grants, McKnow-E, Eureka-Flite2; Belgian Federal Science Policy Office: IUAP P6/04 (Dynamical systems, control and optimization, 2007-2011); EU: ERNSI.

REFERENCES

[1] M. Berry and M. Browne, Algorithms and applications for approximate nonnegative matrix factorization, *submitted for publication*, 2006.

[2] M. Catral, L. Han, M. Neumann and R. Plemmons, On reduced rank nonnegative matrix factorization for symmetric nonnegative matrices, *Linear Algebra and its Applications*, vol. 393 (2004), pp 107-126.

[3] CBCL Face Database 1, MIT Center For Biological and Computation Learning, http://www.ai.mit.edu/projects/cbcl.

[4] L. Finesso and P. Spreij, Nonnegative matrix factorization an I-divergence alternating minimization, *Linear Algebra and its Applications*, vol. 416 (2006), pp 270-287.

[5] G. Golub and C. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, 1996.

[6] D. Ho and P. van Dooren, Nonnegative matrix factorizations with fixed row and column sums, *accepted for publication in Linear Algebra and its Applications*, 2006.

[7] D. Lee and S. Sueng, Learning the parts of object by nonnegative matrix factorization, *Nature*, vol. 401 (1999), pp 788-791.

[8] D. Lee and S. Sueng, Algorithms for nonnegative matrix factorization, *Advances in Neural Information Processing Systems*, vol. 13 (2001), pp 556-562.

[9] V. P. Pauca, J. Piper and R.J. Plemmons, Nonnegative matrix factorization for spectral data analysis, *Linear Algebra and its Applications*, vol. 416 (2006), pp 29-47.

[10] Van den Hof J. *System Theory and System Identification of Compartmental Systems*, PhD Dissertation Rijksuniversiteit Groningen, 1996.