A mixed iteration for nonnegative matrix factorizations

Ştefan M. Şoltuz [a,b,*], B.E. Rhoades [c]

[a] Dawson College, Mathematics Department, 3040 Sherbrooke Street West, Westmount (Montreal), Quebec H3Z 1A4, Canada
[b] Tiberiu Popoviciu Institute of Numerical Analysis, Cluj-Napoca, Romania
[c] Indiana University, Mathematics Department, Bloomingtron, IN, USA

ARTICLE INFO

Keywords:
Non-negative matrix factorization
Lee–Seung iteration

ABSTRACT

We show that, under appropriate conditions, one can create a hybrid between two given iterations which can perform better than either of the original ones. This fact provides a freedom of choice. We also give numerical examples in which we compare our hybrid with the dedicated Lee–Seung iteration.

1. Introduction

1.1. Source separation methods. The non-negative matrix factorization (NMF)

Single-channel source separation problems arise when a number of sources emit signals that are mixed and recorded by a single sensor and one is interested in estimating the original sources of the signals based on the recorded mixture. This problem is ill-posed. Several different model channel source separation methods have been used. One method is the autoregressive model (AR). This model captures temporal correlations at the source, as was shown in [2,3]. In these papers it was proved that, for a single channel mixture of stationary (AR) sources, the (AR) coefficients can be uniquely identified and the sources separated. For non-stationary (AR) sources an adaptive sliding -window was introduced to update the process. A complete study of (AR) models can be found in [4,5].

Let $M(m,n)$ denote the collection of $m \times n$ matrices with nonnegative entries. For a given matrix $V \in M(m,n)$, the Nonnegative Matrix Factorization procedure (*NMF*) is used to find matrices $W \in M(m,r)$ and $H \in M(r,n)$ *such* that $V = WH$. Matrix factorization has many applications. For example it is used in source separation and dimensionality reduction or clustering. In using (*NMF*) it is necessary to compute

$$\arg\min_{W,H} \frac{1}{2} \|V - WH\|^2. \tag{1}$$

An excellent survey on (NMF) and matrix factorization could be found in [15].

Other factorization methods used are Vector Quantization (*VQ*), Principal Component Analysis (*PCA*) and Independent Component Analysis (*ICA*). These can be written in the form $V \approx WH$. The differences between these methods and (*NMF*) are due to the different constraints placed on factoring matrices. In the (*VQ*) method the columns of H are constrained to be unary vectors (i.e., all components are zero except for one element equal to 1). In the (*PCA*) procedure the columns of W and the rows of H must be orthogonal. In the (*ICA*) procedure the rows of H are maximally statistically independent.

A major problem with the (*PCA*) procedure is that it allows the basis vectors to have both positive and negative components and the data are represented by linear combinations of these vectors. In some applications, the presence of negative components contradicts the physical reality. For example, the pixels in a gray scale image must have nonnegative entries, so any image with negative intensities would have no reasonable interpretation. Also STFT magnitude is given by nonnegative quantities. The (*NMF*) procedure was developed in an attempt to address this problem. (See [1,6,7,11,18,19]).

Two iterative (*NMF*) methods are in use – Alternative Least Squares (*ALS*) and that of Lee–Seung (*LS*), the reader may consult [8–13].

1.2. Multiplicative rule. The pointwise product

Consider two functions $F : \mathbb{R}^{M\times N} \to \mathbb{R}$ and $G : \mathbb{R}^{M\times N} \times \mathbb{R}^{M\times N} \to \mathbb{R}$.

Definition 1. [9] One says that $G(H, H')$ is an auxiliary function for $F(H)$ if the conditions

$$G(H, H') \geqslant F(H), \quad G(H, H) = F(H), \tag{2}$$

are satisfied.

In [9] the following quantity is considered as the cost function

$$F(H) = \frac{1}{2}\sum_i \left(V_i - \sum_a W_{ia} H_a\right)^2, \tag{3}$$

where $1 \leqslant i \leqslant m, 1 \leqslant a \leqslant r, V_i$ is the "ith", $(1, n)$ dimension row from V, H_a is the "ath", $(1, n)$ row from H and $H = (H_a)_{1\leqslant a\leqslant r}$.

Lemma 2. [9] *If G is an auxiliary function, then F is nonincreasing under the update*

$$H_{n+1} = \arg\min_H G(H, H_n).$$

The Taylor expansion for F in (3), leads to

$$F(H) = F(H_n) + (H - H_n)\nabla F(H_n) + \frac{1}{2}(H - H_n)^T\left(W_n^T W_n\right)(H - H_n).$$

Leaving both H and H_n as variables one obtains as an auxiliary function for F,

$$G_{Taylor}(H, H_n) = F(H_n) + (H - H_n)\nabla F(H_n) + \frac{1}{2}(H - H_n)^T\left(W_n^T W_n\right)(H - H_n). \tag{4}$$

The quantity $\left(W_n^T W_n\right)$ is a positive semidefinite matrix. Moreover, as noted in [9], the difference matrix between,

$$K_n = diag\left(diag\left(W_n^T W_n H_n^T ./ H_n^T\right)\right) \tag{5}$$

and

$$K_n - \left(W_n^T W_n\right),$$

remains positive semidefinite. (In Matlab, "./" denotes the pointwise division between matrices and "*diag*" the diagonal of a Matrix, seen as vector). Note that we need it twice, in order to keep the dimensionality right. Hence, a similar quantity as the above Taylor expansion, (4),

$$G(H, H_n) = F(H_n) + (H - H_n)\nabla F(H_n) + \frac{1}{2}(H - H_n)^T(K_n)(H - H_n), \tag{6}$$

satisfies the conditions of an auxiliary function of the above F, see [9]. In order to make an upgrade each step for H_{n+1}, the arg min is involved. It is know that under appropriate conditions over Φ, Ψ the quadratic form F, (with Φ a positive defined matrix),

$$F(x) = x^T\Phi x + \Psi x + \Theta, \tag{7}$$

attains its minimum at

$$\bar{x} = \Phi^{-1}\Psi.$$

Thus, by setting (6) into (7),

$$\begin{aligned} \Phi &:= K_n\left(= diag\left(diag\left(W_n^T W_n H_n^T ./ H_n^T\right)\right)\right), \\ \Psi &:= \nabla F(H_n), \\ \Theta &:= F(H_n), \end{aligned}$$

we obtain $\bar{x} = H_{n+1} - H_n$ and therefore the new H_{n+1} is

$$
\begin{aligned}
H_{n+1} - H_n &= \arg\min_H G(H, H_n), \\
H_{n+1} - H_n &= K_n^{-1} \nabla F(H_n), \\
H_{n+1} &= H_n + K_n^{-1} \nabla F(H_n).
\end{aligned} \tag{8}
$$

This leads to the Lee–Seung iterative (multiplicative) method.

1.3. The Lee–Seung multiplicative rule (LS)

More specific, for (8), the matrix H_{n+1} is given by

$$H_{n+1} = H_n - \frac{H_n}{\left(W_n^T W_n H_n\right)}\left(\left(W_n^T W_n H_n\right) - \left(W_n^T V\right)\right).$$

Basically, the rule is to reduce the "distance" (or the cost function) by choosing at each step, appropriate η_n, respectively γ_n,

$$
\begin{aligned}
H_{n+1} &= H_n + \eta_n\left(\left(W_n^T V\right) - \left(W_n^T W_n H_n\right)\right), \\
W_{n+1} &= W_n + \left(\left(V H_{n+1}^T\right) - \left(W_n H_{n+1} H_{n+1}^T\right)\right)\gamma_n.
\end{aligned} \tag{9}
$$

The heart of each iterative method consists in the choice of such η_n and γ_n. Specifically, at step n, in [9], each matrix is given pointwisely by

$$\eta_{ab} = \frac{H_{nab}}{\left(W_n^T W_n H_n\right)_{ab}}, \gamma_{ab} = \frac{W_{nab}}{\left(W_n H_{n+1} H_{n+1}^T\right)_{ab}}, \tag{10}$$

where $(\cdot)_{ab}$, gives the location, (row and column), within the matrix. In (9) set (10). This leads to Hadamard or pointwise multiplication, denoted by "$\cdot$". Eventually, the following iteration method is obtained

$$
\begin{aligned}
H_{n+1} &= H_n \cdot \frac{\left(W_n^T V\right)}{\left(W_n^T W_n H_n\right)}, \\
W_{n+1} &= W_n \cdot \frac{\left(V H_{n+1}^T\right)}{\left(W_n H_{n+1} H_{n+1}^T\right)}.
\end{aligned} \tag{11}
$$

It was reported in [11,13], that the convergence result from [9], actually, does not provide enough conditions for the convergence of (11). A new more stable iteration based on (11)was presented and its convergence was study.

Remark 3. "Lin's modification" for (11), from [11], consists in adding a "small" positive quantity (i.e. $\delta = 10^{-9}$), such that the Lee–Seung iteration becomes:

$$
\begin{aligned}
H_{n+1} &= H_n \cdot \frac{\left(W_n^T V\right)}{\left(W_n^T W_n H_n\right) + \delta}, \\
W_{n+1} &= W_n \cdot \frac{\left(V H_{n+1}^T\right)}{\left(W_n H_{n+1} H_{n+1}^T\right) + \delta}.
\end{aligned} \tag{12}
$$

For this new method, η_n and γ_n, were set to be:

$$\eta_{ab} = \frac{H_{nab}}{\left(W_n^T W_n H_n\right)_{ab} + \delta}, \quad \gamma_{ab} = \frac{W_{nab}}{\left(W_n H_{n+1} H_{n+1}^T\right)_{ab} + \delta},$$

1.4. The Alternative least squares method (ALS)

In (9) set

$$\eta_n = (W_n^T W_n)^{-1},$$

and alternatively,

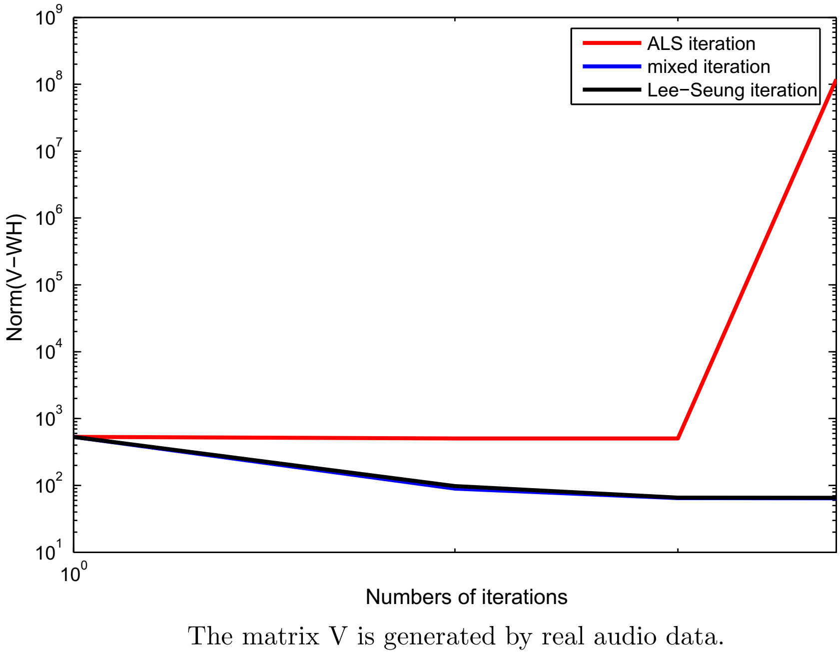

Fig. 1. The matrix V is generated by real audio data.

$$\gamma_n = (H_{n+1}H_{n+1}^T)^{-1},$$

to obtain,

$$H_{n+1} = (W_n^T W_n)^{-1}\left(W_n^T V\right), \quad W_{n+1} = VH_{n+1}(H_{n+1}H_{n+1}^T)^{-1}. \tag{13}$$

The problem with such iteration is that one or both of $(W_n^T W_n)^{-1}$ and $(H_{n+1}H_{n+1}^T)^{-1}$ can be negative. In order to solve this problem one can consider the projection onto the nonnegative orthant, denoted by $P_+[\cdot]$. The above Alternating Least Squares iteration becomes

$$H_{n+1} = P_+\left((W_n^T W_n)^{-1}\left(W_n^T V\right)\right), \quad W_{n+1} = P_+\left(VH_{n+1}(H_{n+1}H_{n+1}^T)^{-1}\right). \tag{14}$$

The convergence for such iteration is described in [7]. We remark that a quantity such as $\left(A^T A\right)^{-1} A$ is similar to the projection operator obtained for the least squares method, (see for example [14]). This method was reported (see also our experiments) to be very fast but unstable, see also [17]. In [17], as well as here, the aim is to obtain a new iteration which has the speed convergence of (ALS) and the stability of (LS).

Typical convergence and divergence behaviors are indicated in Fig. 1. **V** is the spectrogram of an audio signal A3_whistle.wav[1], the frame length of the FFT was set to 512. Hence, the dimension of **V** was $m = 512,\ n = 174$. Both **W** and **H** were randomly initialized. The rank r was set to 7 and 50% overlap between the windows was used for generating the spectrogram. All the three algorithms were applied to decompose the music notes from the audio signal.

1.5. The hybrid method

In (9) insert $\eta = inv(W^{-1}W)$, to obtain the iteration

$$H_{n+1} = H_n + (W_n^T W_n)^{-1}\left(\left(W_n^T V\right) - \left(W_n^T W_n H_n\right)\right). \tag{15}$$

A "general" iteration method for such "H_{n+1}" as in (9) would have the following structure

$$X_{n+1} = X_n + K(Y_n)^{-1}\left(\left(Y_n^T V\right) - \left(Y_n^T Y_n X_n\right)\right),$$
$$Y_{n+1} = Y_n + N(X_{n+1})^{-1}\left(VX_{n+1} - Y_n X_{n+1} X_{n+1}^T\right) \tag{16}$$

or,

$$H_{n+1} = H_n + A(W_n)^{-1}\left(W_n^T V - W_n^T W_n H_n\right)$$
$$W_{n+1} = W_n + B(H_{n+1})^{-1}\left(VH_{n+1} - W_n H_{n+1} H_{n+!}^T\right) \tag{17}$$

where $K(X_n)$ is a surrogate for $A(W_n)$ which may be different at each step. We introduce K to be a nonnegative matrix "close enough" to $A(W_n)$ so as to avoid the errors introduced by using a projection onto the positive orthant. Note that Y_n is different from W_n at each step. But, if Y_n is close enough to W_n, then X_n behaves analogously to $H_{n.}$ We introduced within (16 and (17) two "general" iterations. The study will cover, in this matter, all possible choices for those K, N, A and B within (16 and (17). Appropriate settings for K and N from (16) will lead to (LS) iteration. Eventually we will compare it with (ALS), if A and B are well chosen within (17).

Remark 4. Set $A(W_n)^{-1} = \left(diag\left(diag\left(W_n^T W_n H_n^T./H_n^T\right)\right)\right)^{-1}$ and $B(H_{n+1})^{-1} = \left(\left(diag\left(diag\left(W_n H_{n+1} H_{n+1}^T./W_n\right)\right)\right)\right)^{-1}$ to obtain the Lee–Seung iteration (11). As we can see at each step $A(W_n)$ is changing, even within (15); set $A(W_n)^{-1} = \left(W_n^T W_n\right)^{-1}$ and $B(H_{n+1})^{-1} = \left(H_{n+1} H_{n+1}^T\right)^{-1}$, to obtain the (ALS) iteration.

Our main purposes are the following: first to be able to mix (LS) and (ALS) iterations in order to obtain a better one. Second, we show that "structurally" the two algorithms are not very different and we provide the mathematical background for such hybrid to converge.

2. Main results

2.1. Convergence of the hybrid method

Recall the following Lemma:

Lemma 5. [16] *Let* $\{a_n\}$ *be a nonnegative sequence that satisfies*

$$a_{n+1} \leqslant (1-w)a_n + \sigma_n \mathbf{M},$$

where $w \in (0,1)$ *and* $\mathbf{M} > 0$ *are fixed numbers and* $\{\sigma_n\}$*; is a nonnegative sequence which converges to zero. Then* $\lim_{n\to\infty} a_n = 0$.

The result remains true provided that the coefficient of a_n stays within an interval in $(0,1)$.

Proposition 6. *Let* $\{a_n\}$ *be a nonnegative sequence that satisfies*

$$a_{n+1} \leqslant \lambda_n a_n + \sigma_n \mathbf{M}, \quad \forall n \geqslant n_0,$$

where $\mathbf{M} > 0$ *is fixed number,* $\{\lambda_n\}$ *and* $\{\sigma_n\}$*; are nonnegative sequences such that* $\{\lambda_n\} \subset (0, \Lambda)$*, for some* $\Lambda < 1$ *and* $\lim_{n\to\infty}\sigma_n = 0$*. Then* $\lim_{n\to\infty} a_n = 0$.

Proof. Note that, for each $n \in \mathbb{N}, \lambda_n \leqslant \Lambda$. By defining $(1-w) = \max_n(1-\lambda_n)$, the result follows from Lemma 5. □

Remark 7. If $\lambda_n > 1$ for each n, the Proposition 6 fails. As an example, choose $a_n = n, \lambda_n = 2$ and $\sigma_n = 0$ for each n. Then it follows that $n+1 = a_{n+1} \leqslant \lambda_n a_n + \sigma_n M = 2n$. Proposition 6 remains true if $\lambda_n > 1$, for only a finite subset of $\mathbb{N}$.

For sake of simplicity, through out this paper, we shall consider the sup −norm for all matrices involved. Within Matlab one use "max (max (...))" command.

Theorem 8. *If iteration (17) converges i.e.* $\lim_{n\to\infty} H_n = H^*, \lim_{n\to\infty} W_n = W^*$ *and* $\lim_{n\to\infty} Y_n = W^*$*, and there exists* $\lambda \in (0,1)$ *and* $\mathbf{M} > 0$ *such that for each step* n*, we have the following relations satisfied*

$$\left\| I_{r,r} - K(Y_n)^{-1} W_n^T W_n \right\| \leqslant \lambda < 1,$$
$$\max\left\{\sup_n\left\{\left\|K(Y_n)^{-1}\right\|, \left\|A(W_n)^{-1}\right\|\right\}\right\} \leqslant \mathbf{M}, \tag{18}$$

then iteration (16) is also convergent; i.e. $\lim_{n\to\infty} X_n = H^*$.

Proof. Define $M_n = Y_n - W_n$. Note that,

$$
\begin{aligned}
X_{n+1} &= X_n + K(Y_n)^{-1}\left(Y_n^T V - Y_n^T Y_n X_n\right) \\
&= X_n + K(Y_n)^{-1}\left(\left(W_n^T + M_n^T\right)V - \left(W_n^T + M_n^T\right)(W_n + M_n)X_n\right) \\
&= X_n + K(Y_n)^{-1}\left(W_n^T V - W_n^T W_n X_n\right) \\
&\quad + K(Y_n)^{-1}\left(M_n^T V - \left(M_n^T W_n + W_n^T M_n + M_n^T M_n\right)X_n\right),
\end{aligned} \tag{19}
$$

where W_n is obtained from (17). Using (17) and (19) one obtains

$$
\begin{aligned}
&H_{n+1} - X_{n+1} \\
&= (H_n - X_n) + A(W_n)^{-1}\left(W_n^T V - W_n^T W_n H_n\right) - K(Y_n)^{-1}\left(W_n^T V - W_n^T W_n X_n\right) \\
&\quad - K(Y_n)^{-1}\left(M_n^T V - \left(M_n^T W_n + W_n^T M_n + M_n^T M_n\right)X_n\right) \\
&= (H_n - X_n) + A(W_n)^{-1}\left(W_n^T V - W_n^T W_n H_n\right) - K(Y_n)^{-1}\left(W_n^T V - W_n^T W_n H_n\right) \\
&\quad + K(Y_n)^{-1}\left(W_n^T V - W_n^T W_n H_n\right) - K(Y_n)^{-1}\left(W_n^T V - W_n^T W_n X_n\right) \\
&\quad - K(Y_n)^{-1}\left(M_n^T V - \left(M_n^T W_n + W_n^T M_n + M_n^T M_n\right)X_n\right) \\
&= (H_n - X_n) + K(Y_n)^{-1}\left(W_n^T W_n X_n - W_n^T W_n H_n\right) + \left(A(W_n)^{-1} - K(Y_n)^{-1}\right)\left(W_n^T V - W_n^T W_n H_n\right) \\
&\quad - K(Y_n)^{-1}\left(M_n^T V - \left(M_n^T W_n + W_n^T M_n + M_n^T M_n\right)X_n\right) \\
&= \left(I_{r,r} - K(Y_n)^{-1} W_n^T W_n\right)(H_n - X_n) + \left(A(W_n)^{-1} - K(Y_n)^{-1}\right)\left(W_n^T V - W_n^T W_n H_n\right) \\
&\quad - K(Y_n)^{-1}\left(M_n^T V - \left(M_n^T W_n + W_n^T M_n + M_n^T M_n\right)X_n\right).
\end{aligned}
$$

Thus,

$$
\begin{aligned}
&\|H_{n+1} - X_{n+1}\| \\
&\leqslant \left\|I_{r,r} - K(Y_n)^{-1} W_n^T W_n\right\| \|H_n - X_n\| \\
&+ \left\|A(W_n)^{-1} - K(Y_n)^{-1}\right\| \left\|W_n^T V - W_n^T W_n H_n\right\| \\
&+ \left\|K(Y_n)^{-1}\right\| \|V - W_n X_n\| \left\|M_n^T\right\| \\
&+ \left\|K(Y_n)^{-1}\right\| \left\|W_n^T + M_n^T\right\| \|M_n\| \|X_n\|.
\end{aligned}
$$

We shall consider here the sup – sup-norm such that $\|I_{r,r}\| = 1$. Denote by

$$
\begin{aligned}
a_n &= \|H_n - X_n\|, \\
\lambda_n &= \left\|I_{r,r} - K(Y_n)^{-1} W_n^T W_n\right\|, \\
\sigma_n &= \max\left\{\left\|W_n^T V - W_n^T W_n H_n\right\|, \left\|M_n^T\right\|, \|M_n\|\right\}, \\
\mathbf{M} &= \sup_n \left\{\left\|A(W_n)^{-1} - K(Y_n)^{-1}\right\|, \left\|K(Y_n)^{-1}\right\|, \|X_n\|\right\}.
\end{aligned}
$$

Note that one has $\lambda_n \in (0, \Lambda)$ and $\sigma_n \to 0$, therefore from Proposition 6, we obtain $\lim_{n\to\infty} a_n = \lim_{n\to\infty}\|H_n - X_n\| = 0$, that is $\lim_{n\to\infty}\|X_n\| = H^*$. Using the inequality,

$$
\|X_n - H^*\| \leqslant \|H_n - H^*\| + \|H_n - X_n\|,
$$

it follows that $\lim_{n\to\infty} X_n = H^*$. □

Remark 9. Using Matlab it is easy to verify each step, of the condition of (18) by using (max (max (eye (r,r)-inv (diag (diag ((A'*A*B)./B)))*(A'*A)))), where size (A) = (m,r) and size (B) = (r,n). The second condition of (18), simply demands an upper bound for those matrices involved in the process.

2.2. Further results

In (17), set $A(W_n) = \left(W_n^T W_n\right)^{-1}$, (respectively, $B(H_{n+1}) = \left(H_{n+1}H_{n+1}^T\right)^{-1}$) to obtain the (ALS) method. The above Theorem leads to the following result.

Corollary 10. *If the iteration (15) converges i.e.* ($\lim_{n\to\infty} H_n = H^*, \lim_{n\to\infty} W_n = W^*$ *and* $\lim_{n\to\infty} Y_n = W^*$), *and there exists a* $\lambda \in (0,1)$ *and an* $\mathbf{M} > 0$ *such that for each step* n, *the following relations are satisfied*

$$\left\| I_{r,r} - K(X_n)^{-1} W_n^T W_n \right\| \leqslant \lambda < 1,$$
$$\max\left\{ \sup_n \left\{ \left\| K(X_n)^{-1} \right\|, \left\| \left(W_n^T W_n\right)^{-1} \right\| \right\} \right\} \leqslant \mathbf{M},$$

then the iteration (16) is also convergent; i.e. $\lim_{n\to\infty} X_n = H^*$.

Remark 11.

(a) In other words, this Corollary claims that a hybrid is allowed, provided that appropriate assumptions are satisfied.
(b) Note that by considering the diagonal matrix

$$A(W_n) = diag\left(diag\left(W_n^T W_n H_n^T./H_n^T\right)\right),$$

(17) becomes the Lee–Seung iteration (with the Hadamard product).

Proposition 12. *In Theorem 8 one can replace*

$$\left\| I_{r,r} - K(Y_n)^{-1} W_n^T W_n \right\| \leqslant \lambda,$$

with

$$\frac{1-\lambda}{\left\| W_n^T W_n \right\|} \leqslant \left\| K(Y_n)^{-1} \right\|.$$

Proof. Note that

$$1 - \left\| K(Y_n)^{-1} \right\| \left\| W_n^T W_n \right\| \leqslant 1 - \left\| K(Y_n)^{-1} W_n^T W_n \right\|$$
$$\leqslant \left\| I_{r,r} - K(Y_n)^{-1} W_n^T W_n \right\| \leqslant \lambda,$$

to obtain the conclusion. □

By duality, one can consider the case in which $\lim_{n\to\infty} X_n = H^*$ to obtain.

Corollary 13. *If the iteration (17) converges (i.e.* $\lim_{n\to\infty} H_n = H^*, \lim_{n\to\infty} W_n = W^*$ *and* $\lim_{n\to\infty} X_n = H^*$), *and there exists a* $\lambda \in (0,1)$ *and an* $\mathbf{M} > 0$ *such that for each step* n, *we have the following relations satisfied*

$$\left\| I_{r,r} - H_{n+1} H_{n+1}^T N(X_n)^{-1} \right\| \leqslant \lambda < 1,$$
$$\max\left\{ \sup_n \left\{ \left\| N(X_n)^{-1} \right\|, \left\| B(H_{n+1})^{-1} \right\| \right\} \right\} \leqslant \mathbf{M},$$

then the iteration (16) is also convergent; i.e. $\lim_{n\to\infty} Y_n = W^*$.

As in [9], the next step is to consider the second part of (9) with $B(H_{n+1}) = (H_{n+1}H_{n+1}^T)$; i.e.

$$W_{n+1} = W_n + (V - W_n H_{n+1}) H_{n+1}^T (H_{n+1} H_{n+1}^T)^{-1},$$

and the following quantity from (16),

$$Y_{n+1} = Y_n + (V - Y_n X_{n+1}) X_{n+1} N(X_n)^{-1},$$

to obtain a similar result to Corollary 10.

Corollary 14. *If the iteration (15) converges (i.e.* $\lim_{n\to\infty} H_n = H^*, \lim_{n\to\infty} W_n = W^*$ *and* $\lim_{n\to\infty} X_n = H^*$) *and there exists a* $\lambda \in (0,1)$ *and an* $\mathbf{M} > 0$ *such that for each step* n, *we have the following relations satisfied*

$$\left\|I_{r,r} - H_{n+1}H_{n+1}^T N(X_n)^{-1}\right\| \leqslant \lambda < 1,$$
$$\max\left\{\sup_n\left\{\left\|N(X_n)^{-1}\right\|, \left\|\left(H_{n+1}H_{n+1}^T\right)^{-1}\right\|\right\}\right\} \leqslant \mathbf{M},$$

then the iteration (16) is also convergent; i.e. $\lim_{n\to\infty} Y_n = W^*$.

A result similar to Proposition 12 also holds. For practitioners the changed condition may be more useful.

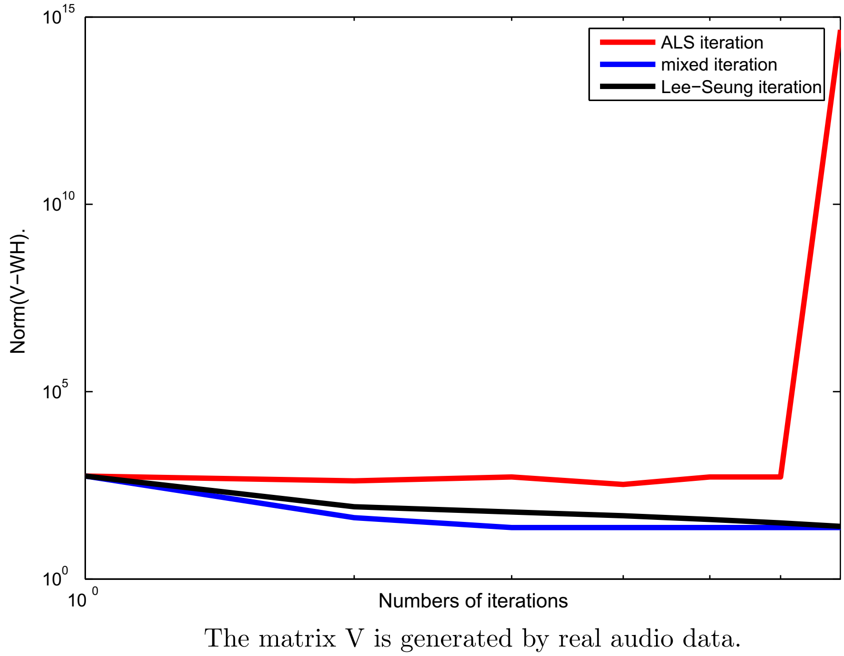

Fig. 2. The matrix V is generated by real audio data.

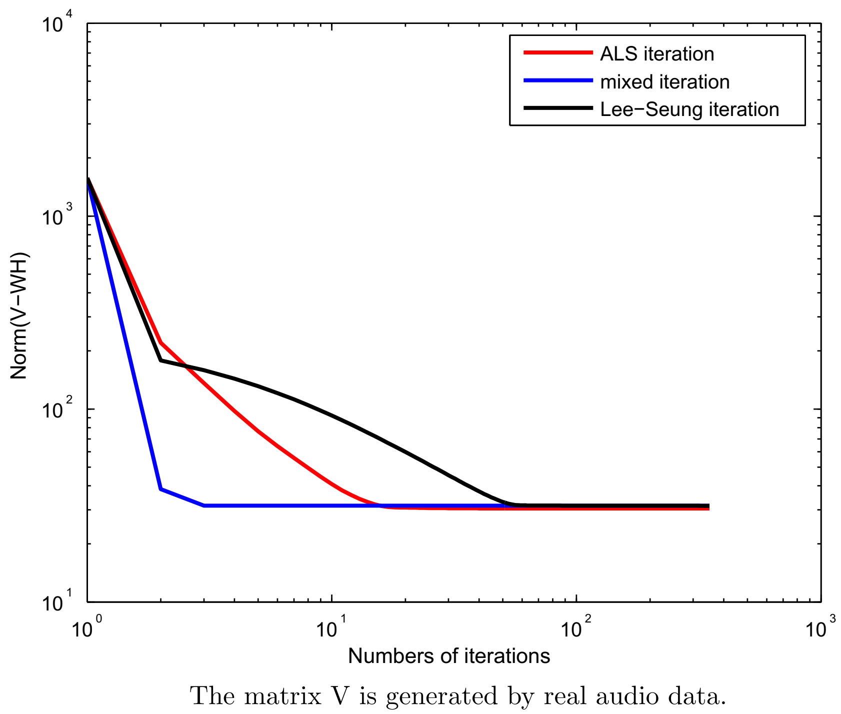

Fig. 3. The matrix V is generated by real audio data.

Remark 15. Analogously, one can replace at each step

$$\left\| I_{r,r} - H_{n+1}H_{n+1}^T N(X_n)^{-1} \right\| \leqslant \lambda,$$

by

$$\frac{1-\lambda}{\left\| H_{n+1}H_{n+1}^T \right\|} \leqslant \left\| N(X_n)^{-1} \right\|.$$

3. Numerical examples. Convergence and divergence behaviors

In the first experiment we let V denote the spectrogram of the audio signal C6_frenchhorn.wav[2], where the frame length of the FFT was set at 512. Thus the dimension of V was $m = 512$ and $p = 174$. Both **W** and **H** were randomly initialized. The rank r was set to 7 and 50% overlap between the windows for generating the spectrogram. All three algorithms were applied to decompose the music notes from the audio signal. The convergence curves were averaged over 20 independent tests. The results are shown in Fig. 2.

In our second experiment, we generated **V** synthetically as the absolute value of a zero-mean Gaussian distributed random variable and initialized **W** and **H** in the same way. The dimensions of these matrices were set as $m = 500, n = 300$ and $r = 7$. The Matlab Program performed 20 independent random tests in which both **W** and **H** were kept the same for all the three algorithms. The evolution of the cost function averaged over the 20 tests. In Fig. 3 are shown the behaviors of the proposed algorithm, as well as the (LS) and (ALS) algorithms. When r is increased to 13 or higher, the (ALS) algorithm becomes unstable, while the proposed algorithm still converges, even though its rate of convergence becomes slower than that of the (LS) algorithm.

Acknowledgement

The first author is indebted to Ioana C. Soltuz, Maria Soltuz for their constant support throughout this journey we made together. Also, the authors are indebted to a referee for carefully reading the paper and for making useful suggestions.

References

[1] R. Albright, J. Cox, D. Duling, A. Langville, C. Meyer, Algorithms, initializations and convergence for the nonnegative matrix factorization, NCSU Technical Report Math 81706, 2006, submitted for publication, URL <http://meyer.math.ncsu.edu/Meyer/Abstracts/Publications.html>.

[2] R. Balan, J. Rosca, A spectral power factorization, Siemens Corporate Research. Princeton, NJ, Tech. Rep. SCR-01-TR-703, Sep 2001.

[3] R. Balan, A. Jourjine, J. Rosca, AR process and sources can be reconstructed from degenerate mixtures, in Independent Component Analysis and Blind Signal Separation, International Conference on (ICA), Jan 1999, pp. 467–472.

[4] P.J. Brockwell, R. Davis, Time Series: Theory and Methods, Springer, 1991.

[5] P.J. Brockwell, R. Davis, Introduction to Time Series and Forecasting, Springer, 1996.

[6] A. Ben Hamza, D.J. Brady, Reconstructin of reflectance spectra using robust NMF, IEEE Trans. Signal Process. 54 (2006) 9. de unde am ciatt.

[7] M. Berry, M. Browne, A. Langville, P. Pauca, R.J. Plemmons, Algorithms and appliactions for approximation nonnegative matrix factorization, Computational Statistics and Data Analysis, 2006.

[8] D. Donoho, V. Stodden, When does non-negative matrix factorization give a correct decomposition into parts?, in: Advances in Neural Information Processing Systems (NIPS), vol. 17.

[9] D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, Adv. Neural Inf. Process. Syst. 13 (2) (2005) 556–562.

[10] Daniel D. Lee, H. Sebastian Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (6755) (1999) 788–791.

[11] C.J. Lin, On the convergence of multiplicative update algorithms for nonnegative matrix factorization, IEEE Trans. Neutral Networks 18 (6) (2007) 1589–1596.

[12] C.J. Lin, Projected gradient methods for non-negative matrix factorization, Neural Comput. (2007). to be published.

[13] E.F. Gonzales, Y. Zhang, Accelerating the Lee–Seung algorithm for non-negative matrix factorization, Dept. Comput. Appl. Math. Rice Univ. Houston TX, Tech. Rep. 2005.

[14] G. Strang, Introduction to Applied Mathematics, Wellesley-Cambridge Press, 1986.

[15] M.N. Schmidt, Single-channel source separation using non-negative matrix factorization, Ph. D. Thesis, Technical University of Denmark, 2008.

[16] Ştefan M. Şoltuz, Sequences supplied by inequalities and applications, Revue d'analyse numerique et de theorie de l'approximation 29 (2) (2001) 207–212.

[17] Ştefan M. Şoltuz, W.Wang, P. Jackson, A hybrid iterative algorithm for nonnegative matrix factorization, Workshop 15th on Statistical Signal Processing, 2009. SSP '09. IEEE/SP Cardiff.

[18] W. Wang, X. Zou, Nonnegative matrix factorization based on projected nonlinear conjugate gradient algorithm, ICARN 2008.

[19] R. Zdenuk, A. Cichocki, Nonneagtive matrix factorization with quadratic programming, Neurocomputing 71 (2007) 2309–2320.

Available at www.ee.surrey.ac.uk/Personal/W.Wang/demondata.html